NASA Graduate Student Researchers Program Final Report

Grant Number NGT-1-02005

By Nicholas K. Borer
Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332-0150

June 2005

Decomposition-Based Decision Making for Aerospace Vehicle Design

Research Summary for NASA Graduate Student Researchers Program Final Report

By Nicholas K. Borer

Decomposition-Based Decision Making for Aerospace Vehicle Design

Thesis Advisory Committee

Dr. Dimitri Mavris, GT/AE, Thesis Advisor

Dr. Daniel Schrage, GT/AE

Dr. Alan Wilhite, GT/AE, National Institute of Aerospace

Mr. Craig Nickol, NASA Langley Research Center

Abstract

Most practical engineering systems design problems have multiple and conflicting objectives. Furthermore, the satisfactory attainment level for each objective ("requirement") is likely uncertain early in the design process. Systems with long design cycle times will exhibit more of this uncertainty throughout the design process. This is further complicated if the system is expected to perform for a relatively long period of time, as now it will need to grow as new requirements are identified and new technologies are introduced. These points identify a need for a systems design technique that enables decision making amongst multiple objectives in the presence of uncertainty.

Traditional design techniques deal with a single objective or a small number of objectives that are often aggregates of the overarching goals sought through the generation of a new system. Other requirements, although uncertain, are viewed as static constraints to this single or multiple objective optimization problem. With either of these formulations, enabling tradeoffs between the requirements, objectives, or combinations thereof is a slow, serial process that becomes increasingly complex as more criteria are added.

This research proposal outlines a technique that attempts to address these and other idiosyncrasies associated with modern aerospace systems design. The proposed formulation first recasts systems design into a multiple criteria decision making problem. The now multiple objectives are decomposed to discover the *critical characteristics* of the objective space. Tradeoffs between the objectives are considered amongst these critical characteristics by comparison to a probabilistic *ideal tradeoff* solution.

The proposed formulation represents a radical departure from traditional methods. A pitfall of this technique is in the validation of the solution: in a multi-objective sense, how can a decision maker justify a choice between non-dominated alternatives? A series of examples help the reader to observe how this technique can be applied to aerospace systems design and compare the results of this so-called *Decomposition-Based Decision Making* to more traditional design approaches.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Most practical engineering systems design problems have multiple and conflicting objectives. Furthermore, the satisfactory attainment level for each objective ("requirement") is likely uncertain early in the design process. Systems with long design cycle times will exhibit more of this uncertainty throughout the design process. This is further complicated if the system is expected to perform for a relatively long period of time, as now it will need to grow as new requirements are identified and new technologies are introduced. These points identify a need for a systems design technique that enables decision making amongst multiple objectives in the presence of uncertainty.

Traditional design techniques deal with a single objective or only a few objectives that are often aggregates of the overarching goals sought through the generation of a new system. Other requirements, although uncertain, are viewed as static constraints to this single- or multi-objective optimization problem. With this formulation, enabling tradeoffs amongst the requirements, objectives, or combinations thereof is a slow, serial process that becomes increasingly complex as more criteria are added.

The research in this document outlines a technique that attempts to address these and other idiosyncracies associated with modern aerospace systems design. The proposed formulation first recasts systems design into a fully multiple criteria decision making problem. The now multiple objectives are decomposed to discover the *critical characteristics* of the objective space. Tradeoffs amongst the objectives are considered amongst these critical characteristics by comparison to a probabilistic *ideal tradeoff* solution.

The proposed formulation represents a radical departure from traditional methods. A pitfall of this technique is in the validation of the solution: in a multi-objective sense, how can a decision maker justify a choice between non-dominated alternatives? A series of examples help the reader to get a feel for how this technique can be applied to aerospace systems design and compare the results of this so-called *Decomposition-Based Decision Making* to more traditional design approaches.

1.1 Motivation

The motivation for this research began with the author's involvement in a first-year graduate design competition involving a multi-mission aircraft. This competition required the design team to determine the salient characteristics of a vehicle with multiple, conflicting mission requirements and subsequently to design an aircraft to these features. The identification of these features followed an analysis of the vehicle's "mission space" to determine which requirements would be the design drivers [Mavris and Borer, 2001].

This first approach at multi-mission sizing represented a technique that is most typical of modern multi-mission vehicle design; that is, to design the aircraft to only the most stringent requirements. Though effective, it does little to accommodate tradeoffs and can quickly result in an infeasible design. If nothing else, this initial design project broached two subject areas critical to the development of a new technique: the need for a multi-mission sizing method and for the ability to capture uncertainty in requirements specification.

1

1.1.1 Multi-Mission Sizing

The first 11 seconds of manned heavier-than-air flight began on 17 December 1903, now over a century ago. Since those first flights off the dunes of Kill Devil Hills, aircraft have evolved into massive, complex, and versatile machines. The first practical use of aircraft for carrying passengers or cargo was realized within a decade, and its military applications for observation and attack soon after. By the fourth decade of powered flight aircraft were used for a variety of missions: cargo and troop transport, precision bombing, aerial photography, escort, interception, and many others. By the end of the sixth decade of powered flight manufacturers were building aircraft for every conceivable mission.

However, as the number of missions for aircraft increased, manufacturers and operators found that production and operations costs increased and tactical redundancy suffered. This problem was compounded as advances in technology required a wider array of missions, such as supersonic attack, airborne early warning, and electronic surveillance. Certain aircraft were tasked with synergistic missions in an effort to alleviate these problems.

The notion of the multi-role aircraft, though not new, has seen increased emphasis over the years. This trend, though difficult to explicitly document, can be exemplified in the composition of aircraft carrier air wings. Figure 1 compares the number of different types of aircraft aboard an aircraft carrier versus the number of missions performed by these aircraft for different historical periods, compiled from several different resources [Department of the Navy, 2005; Federation of American Scientists, 2005; Toppan, 2003].
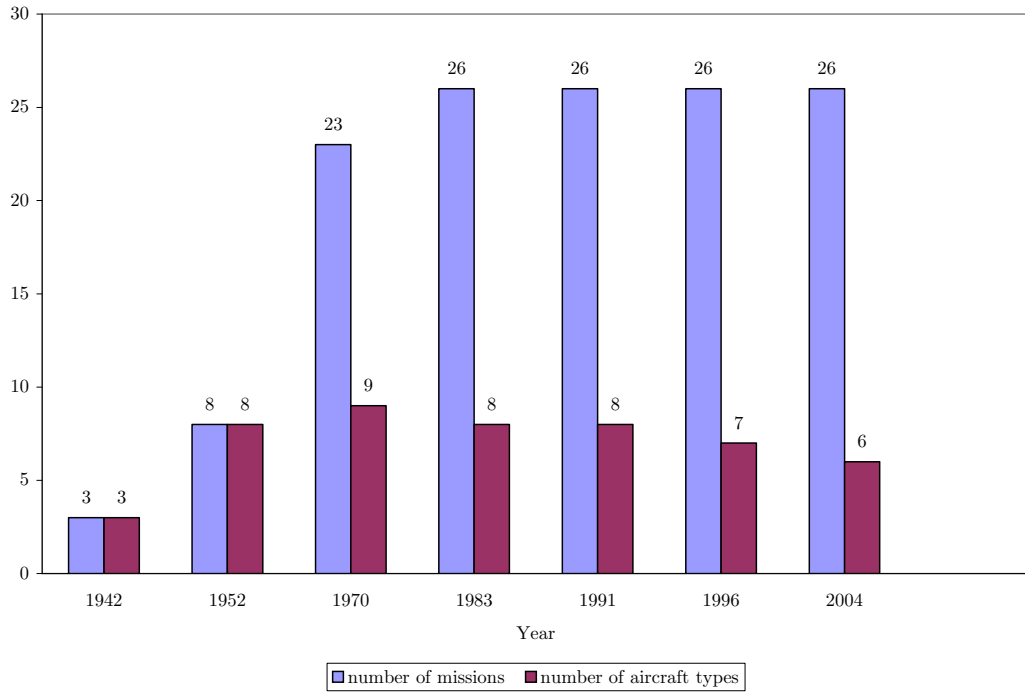
Figure 1: Carrier Air Wing Composition

This figure shows that the number of missions has climbed rapidly and levelled off with respect to aircraft carrier operations, yet the number of different aircraft types saw a short increase followed by a steady decrease. The carrier air wing of the future will deploy even

fewer craft as new programs such as the multi-mission capable F/A-18E and F-35C aircraft enter service and replace aging airframes [Young et al., 1998; Sherman and Hardiman, 2003].

The increased emphasis of multi-role capability for aircraft can be traced to two principal reasons: affordability and redundancy. Employing a single aircraft type or variant of said type for several roles reduces the need for specialized equipment, personnel, training, and spare parts, among other resources. It also enables economies of scale through larger-scale mass-production of the same airframe. All of these attributes point to a reduction in overall life-cycle costs at tactical (and higher) levels. Redundancy is increased for many of the same reasons. As an example, consider two squadrons: one with 10 dedicated attack and 10 dedicated fighter aircraft, and one with 20 multi-role fighters. If five aircraft are lost on a fighter mission, the squadron with dedicated aircraft exhibits a 50% loss in fighter capability, while the multi-role squadron can spread that loss out evenly over its attack and fighter capability as needed. This is important not only in military applications but also civil applications as many passenger air carriers begin to investigate the feasibility of intermodal (passenger by day, cargo by night) transports [Nelson et al., 2001].

Unfortunately, multi-mission capability comes at a price. One maxim in aerospace design is that *every additional requirement imposed on a system will compromise the design in some way.* This concept of "no free lunch" will manifest itself as a decrease in effectiveness in another requirement, such as a performance or vehicle-level cost measure. The only exception to this rule is in the trivial case of a completely dominated or redundant requirement, in which case the true multi-role capability of the system is not expanded.

1.1.2 Requirements Uncertainty

Successful design of any system begins with specification of requirements. These requirements are composed of one or more objectives and zero to many constraints. As aerospace vehicle design has progressed, the requirements imposed on vehicles have become more numerous and stringent. Requirements are influenced by advances in technology, the environment, politics, and many other factors.

Most engineering problem solving techniques begin with a static set of requirements. Any change to these requirements can change what was once the best solution to a different solution. As design cycle times increase, there is a larger chance that the various factors shaping the requirements will change as well, thus resulting in changes to the original requirements. This can have a substantial effect on modern aerospace vehicle design. To compare design cycle times, consider that the Lockheed P-80 Shooting Star, the first operational jet fighter made in the United States, went from drawing board to first flight in 143 days [Rich and Janos, 1994]. Compare this to a modern jet fighter, the F-22 Raptor. The Raptor itself was conceived in 1983 and first flew in 1997, and as of this writing in 2005 the first squadron is becoming operational. This example, though radical, exemplifies the increase in design cycle time: 143 days to 14 years! Even in this vein, one must consider that the original requirements for the F-22 date back to the Advanced Tactical Fighter concept first proposed in 1971 [Piccirillo, 1998]. Over that time numerous changes have been seen in the technological environment (stealth), political environment (the fall of the USSR), and the fiscal environment, to name a few.

Further complicating requirements specification is the concept of growth potential. Aerospace vehicles, at their extraordinary expense and complexity, are expected to perform for a relatively long period of time. Over this time the operational environment and associated requirements will change. Commercial operators will be faced with a different

market, more stringent environmental regulations, and safety regulations. Military operators will face new enemies and weapons systems. Overall, a successful aircraft will need to have the ability to grow and change throughout its life cycle. Compare the service of the B-52 subsonic strategic bomber to that of the B-58 supersonic bomber: both were contemporaries; the earliest operational B-52 first flew in 1955 and the B-58 in 1960. The latest models of both aircraft were produced in 1962. However, the B-52 is so effective, affordable, and adaptable for strategic bombing that it is planned to remain in service until 2040, 85 years after the first model and 78 years after the last of the current models entered service! The B-58 only lasted until 1970, 10 years after entering service simply because it was not nearly as versatile or adaptable in its future environment [Baugher, 2005; Federation of American Scientists, 2005].

These all point to another maxim related to aerospace vehicle design: *The requirements specified for the system today will be different than those faced at the beginning of its operational life, which will further change throughout the system's entire life cycle.*

CHAPTER II

REQUIREMENTS AND AIRCRAFT SIZING

The bases of multi-mission sizing have to do with the broader spectrum of systems design methods and requirements specification. Therefore, to gain an appreciation of the issues associated with multi-mission sizing one must first understand the idiosyncracies associated with the engineering design process, requirements specification, and vehicle sizing methodologies. Each of these areas has evolved over the years to take advantage of a larger body of knowledge and better resources to tackle these various problems. What follows is an overview of the origins and growth experienced in these fields, though the reader is cautioned that this is by no means a comprehensive review.

2.1   The Engineering Design Process

Several models exist for the engineering design process related to aerospace vehicles. Raymer suggests a serial, three-tiered process of conceptual design, preliminary design, and detail design, with requirements feeding into the conceptual design process and fabrication following detail design [Raymer, 1999]. This text, meant as a single source for undergraduate engineering students, presents a deliberately simplified process. More rigorous techniques involve more stages, such as detailed requirements generation studies and considerations of the entire life cycle of the vehicle. Asimow enumerates an eight-phase design process to embody all of these concepts, from feasibility studies to planning for retirement [Asimow, 1962]. Dieter revives and modernizes these concepts in his text on systems design [Dieter, 2000]. Of particular interest are the initial phases of what Dieter refers to as conceptual design and embodiment design. This refined design process is illustrated in Figure 2.

The life-cycle approach to engineering design is imbedded into the concept of systems engineering. The Defense Systems Management College [Leonard, 1999] defines systems engineering as:

> . . . an interdisciplinary engineering management process to evolve and verify an integrated, life cycle balanced set of system solutions that satisfy customer needs.

The text further discusses four phases in the development of a design. The process first starts with concept studies, resulting in a *functional baseline*. This baseline is used in the system definition phase until an *allocated baseline* is defined, which carries over to the preliminary design stage. Here, the process continues until the definition of a *product baseline*, used in the final detail design phases. In general, these phases follow the same general approach as those reference by Raymer and Dieter above (requirements, conceptual design / system definition, preliminary / embodiment design, detailed design). However, the Department of Defense greatly elaborates on methods for identifying requirements for systems engineering. The core process involves requirements analysis, functional analysis, system synthesis, and system analysis and control. This process is illustrated in Figure 3.

Conceptual Design

Define Problem	Gather Information	Concept Generation	Evaluation of Concepts
Problem statement	Internet	Brainstorming	Pugh concept selection
Benchmarking	Patents	Functional decomposition	Decision matrices
QFD	Trade literature	Morphological chart	
PDS			
Project planning			

Conceptual Design

Embodiment Design

Product Architecture	Configuration Design	Parametric Design	Detail Design
Arrangement of physical elements to carry out function	Preliminary selection of materials and manufacturing process	Robust design	Detailed drawings and specifications
	Modeling / sizing of parts	Tolerances	
		Final dimensions	
		DFM	

Embodiment Design

Figure 2: The Engineering Design Process [Dieter, 2000]

Process Input

- Customer needs / objectives / requirements
  - Missions
  - Measures of effectiveness
  - Environments
  - Constraints
- Technology base
- Output requirements from prior development effort
- Program decision requirements
- Requirements applied through specifications and standards

Requirements Analysis

- Analyze missions and environments
- Identify functional requirements
- Define / refine / performance and design constraint requirements

Requirements Loop

Functional Analysis / Allocation

- Decompose to lower-level functions
- Allocate performance and other limiting requirements to all functional levels
- Define / refine functional interfaces (internal / external)
- Define / refine / integrate functional architecture

Design Loop

Synthesis

- Transform architectures (functional to physical)
- Define alternative system concepts, configuration items, and system elements
- Select preferred product and process solutions
- Define / refine physical interfaces (internal / external)

Verification

System Analysis and Control (Balance)

- Trade-off studies
- Effectiveness analyses
- Risk management
- Configuration management
- Interface management
- Data management
- Performance Measurement
  - SEMS
  - TPM
  - Technical reviews

Related Terms:

Customer = Organizations responsible for primary functions

Primary Functions = Development, production / construction, verification, deployment, operations, support, training, disposal

System Elements = Hardware, software, personnel, facilities, data, material, services, techniques

Process Output

- Development level dependent
  - Decision database
  - System / configuration item architecture
  - Specifications and baselines

Figure 3: The Systems Engineering Process [Leonard, 1999]

6

The systems engineering approach is among the first to identify the need to include tradeoffs of functional requirements. However, as discussed later, the overall system requirements ("measures of effectiveness" in Figure 3) remain largely static after the initial requirements specification stage.

2.1.1 Requirements Specification

The objective of engineering design is to produce a product (mechanical or otherwise) capable of satisfying a need. This can be a new need brought about by scarcity, an improvement to an existing need brought about by technology, and any other number of cases. Alone, these needs are generally vague and do not necessarily imply a solution. The engineer then has a variety of tools available to translate these needs, "the voice of the customer," into technical requirements, "the voice of the engineer." Often in order to do this a few solution configurations are necessary, though the specific parameters of said configurations need not be determined. For example, there is a need to transport people across the Atlantic Ocean. There are a huge number of solutions available, however impractical: by sea, by air, by a giant bridge, by space, and more. With the configurations narrowed down, one may begin to create technical requirements. If the solution is an aircraft, what range should it be capable of? How many passengers should it carry? What safety features must be incorporated? These and other questions form the basis of the generation of top-level requirements.

For small systems, requirements specification is relatively straightforward. Often there are a few constraints and one objective. These requirements (constraints and objective function) follow from a relatively simple set of rules - do not exceed the maximum stress, do not buckle under load, minimize cost of manufacture, etc. However, more complex systems involve multiple requirements that may be difficult to characterize. Thus, numerous methods have been developed within systems engineering to handle the translation of "needs" into "requirements." The needs may be characterized through items such as market surveys and customer questionnaires. This "voice of the customer" can be mapped to the "voice of the engineer" through Quality Function Deployment (QFD). This technique uses a series of "House of Quality" worksheets to determine relevant engineering requirements [Dieter, 2000]. These represent but a few options available when determining overall engineering requirements. Several other techniques used in systems engineering around outlined by Brassard [Brassard and Ritter, 1994].

Requirements specification techniques have evolved in step with design methods. Aircraft sizing, to be covered in greater detail in the subsections that follow, has advanced substantially as the statistical database and computer power have grown. As such, those setting the requirements are able to see the physical effects of "pushing the envelope" when specifying what it is a vehicle can do [Czysz et al., 1973; Parker, 1986; Gerhards et al., 2000]. Advanced systems engineering techniques have evolved such that requirements specification and systems design have become codependent [Schrage, 1999].

The design of aircraft often begins with the specification of several specific requirements. Some of the most basic requirements are related to the missions the aircraft is expected to perform. These missions are in fact nominal abstractions of reality; a "slice" of the vehicle's overall mission capability. These mission requirements are added to the aircraft's performance, economic, and operational needs to create an overall requirements set. Some methods to design or "size" an aircraft to these requirements are expanded upon below.

2.2   Traditional Single-Objective Approaches

The earliest approaches to systematic aircraft sizing revolved around the minimization of a single objective. This objective had to be an aggregate measure of development, production, and operating cost. It could be minimized subject to critical performance parameters borne of the mission the vehicle was expected to accomplish. This would help the engineer determine the most "affordable" vehicle, where "affordability" is thought of as ratio of performance to life-cycle cost. The wealth of data and relatively homogenous vehicle designs post-World War II provided engineers with the data to provide such an objective, and it was found that Takeoff Gross Weight (TOGW or GW) seemed to vary linearly with most cost metrics. The era of single-objective sizing was born.

One problem with single-objective optimization at the time was the lack of reliable numerical methods (or the machines to work them) for optimization. Thus, most techniques were graphical. The first sizing methods were therefore graphical as well. These first systems engineering techniques sought to bring together the multiple disciplines related to aircraft sizing into a few parametric "scaling laws" related to TOGW. Hiller Helicopters published their R_f method in the mid-1950s as a graphical technique for minimum gross-weight estimation [Joy and Simonds, 1956; Simonds, 1956]. In this work, R_f referred to the various scaling laws for the fraction of fuel weight to gross weight. These were mostly statistical regressions with respect to dimensional parameters, environmental parameters, other constants, and, of course, gross weight. In this work, the idea was to graph the curves of fuel required and fuel available. The point where they met would therefore be the right "size" to best meet the requirements, hence the moniker "sizing." The R_f method would further be used for minimization of gross weight by plotting several fuel required and fuel available curves for different settings of critical sizing variables such as thrust-to-weight ratio and wing loading (or disk loading for rotorcraft).

The purely graphical methods began to give way to numerical schemes combined with graphical aids as the availability of numerical methods (and machines to handle these computations) increased. Most notably, the fuel-balance technique evolved. This was a numerical scheme used to automatically evaluate vehicle scaling laws and find the gross weight where fuel required and fuel available met, hence, "fuel balance." This age also saw the increased emphasis on *synthesis*, the act of bringing together multiple disciplinary analyses into a single, integrated scheme. This enabled the initial scaling laws to expand to include more sophisticated terms for aerodynamics, propulsion, weight, structures, cost, and others. This in turn made more design variables available to the systems engineer while optimizing the vehicle to its requirements. Now the design engineer could make decisions from carpet plots of weight versus performance metrics. Some basic fuel-balance parametric sizing and synthesis techniques are illustrated by Pugliese [Pugliese, 1971].

In the years since, the statistical database for aircraft has increased substantially, allowing for more accurate and detailed scaling laws. Engineers sought to create better fits through more detailed, computer-assisted statistical techniques as well as by classifying various scaling laws for different classes of vehicles [Greenway and Koob, 1978]. Now "basic" techniques taught to today's undergraduate aerospace engineers include the tools provided by Roskam [Roskam, 1989], Mattingly [Mattingly et al., 2002], and Torenbeek [Torenbeek, 1982], to name a few. These techniques continue to work well for sizing single-mission aircraft, especially those that fall within the realm of vehicle classes described thus far. However, these methods are becoming increasingly limited as engineers seek revolutionary concepts and configurations that are outside of the statistical database.

2.3   Modern Sizing Methods

The further increases in computational power over the years prompted the development of several stand-alone aircraft sizing codes. These monolithic codes are often semi-empirical, relying on statistical data and physics models that range from crude to highly sophisticated. Some examples of highly developed sizing and synthesis codes include NASA's FLight Optimization System (FLOPS) [McCullers, 1984] and AirCraft SYNThesis (AC-SYNT) [Myklebust and Gelhausen, 1994] programs. Furthermore, these monolithic codes allow an engineer to track more than just gross weight; rather, they can have almost instant computation of other cost and performance metrics for more informed decision making and tradeoff analysis [Eckels, 1983; Simos and Jenkinson, 1986].

As advanced as they are, monolithic codes still suffer a lack of flexibility. Often, the codes contain assumptions that limit the diversity of configurations that can be investigated. Other times, in the name of generality or computational efficiency, these codes contain deliberately simplified analyses. In either case, they may not be appropriate on their own for the investigation of radical concepts or for higher-fidelity design exploration. However, they often have "handles" built in to allow for higher-fidelity results to be directly input into the system. This has allowed for the next evolution in systems synthesis and design: the integrated environment.

An integrated environment is a collection of disciplinary analyses capable of direct communication with one another. However, analysis codes use a variety of input and output formats, so it can be difficult to get the various codes to communicate effectively with any flexibility. Recent years have seen the development of environments capable of "wrapping" each code in such a way that inputs and outputs can be readily swapped amongst the codes. These codes (and wrappers) are available on a common server available to multiple workstations so that several engineers can access and change the data and run the sizing programs as needed. Some environments used include Phoenix Integration's *ModelCenter* [Phoenix Integration, 2005] and Engineous Software's *iSIGHT* [Engineous Software, 2005]. These environments provide wrappers for a variety of programs, allow users to create custom wrappers, and give the user control over inputs and output parsing. They have been used to great effect as illustrated in a case study by Lockheed Martin [Carty, 2002]. Current developments in integrated environments cite the use of specialized formats that can be used without a central server, instead relying on distributed networks or the internet via XML protocols [Lin and Afjeh, 2002; Kam and Gage, 2003].

2.3.1   Multidisciplinary Design Optimization and Statistical Techniques

The availability of a suitable integrated environment gives the engineer enormous flexibility in the range of concepts that may be investigated. However, once this environment is available, dozens of systems engineering, decision making, and optimization techniques are available to further assist the engineer. Perhaps the most fundamental is the Design Structure Matrix or n^2 diagram [AIAA Technical Activities Committee, 1991]. This tool allows for the analyses within the integrated environment to be moved in such a way as to eliminate as many feedbacks as possible (where iteration is present) and to allow analyses to be executed in parallel to save overall system evaluation time. An example of this is presented in Figure 4. This formulation also opens up many different Multidisciplinary Design Optimization (MDO) tools such as Collaborative Optimization [Braun et al., 1996; Sobieszczanski-Sobieski et al., 1998], Optimizer Based Decomposition [Ledsinger and Olds,

1998], Systems Sensitivity Analysis [Sobieszczanski-Sobieski, 1990; Olds, 1994], and others.
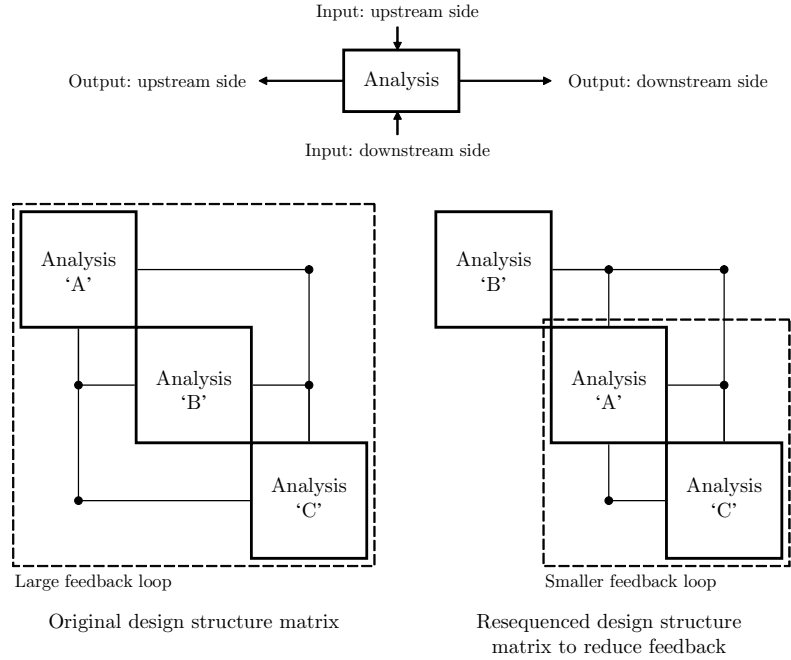
Figure 4: Sample Design Structure Matrix Before and After Restructuring

One drawback of working directly with the integrated environment is that the overall system evaluation time may be relatively long due to the execution times of the individual analyses, in series or parallel. Even modern MDO methods may not enable an engineer to rapidly investigate the entire design space available to a given concept. Therefore, high-fidelity approximations and statistical techniques can be used to speed up system execution time and explore every concept available within the design space. Three principle methods are capable of this: Monte Carlo Simulation (MCS) of the design space combined with the original integrated environment, MCS of the design space combined with a surrogate model (approximation) of the original environment, or Fast Probability Integration (FPI) of the design space combined with the original environment [Fox, 1994; Mavris and Bandte, 1997]. The first is the highest fidelity solution but takes the most time; the latter two provide time savings at reductions in fidelity. All of these techniques rely on the generation of Cumulative Distribution Functions (CDFs) to determine the portion of the design space that can satisfy the specified requirements. These CDFs can be viewed individually to determine the probability of success for a given metric. However, this does not determine the probability of meeting more than just one particular metric. Bandte's Joint Probability Decision Making (JPDM) technique combines the Probability Density Functions (PDFs) generated during probabilistic design to determine the portion of the design space that can meet more than one requirement [Bandte, 2000]. An illustration of a single CDF and a joint distribution function is given in Figure 5.

Several studies have been made using MCS with polynomial approximations [Mavris and Bandte, 1995; DeLaurentis et al., 1996], as well as MCS with FPI [Mavris et al., 1997]. A popular method for approximation of relatively well-behaved systems is Response Surface Methodology (RSM) [Neter et al., 1996] because it is capable of capturing linear, non-linear,
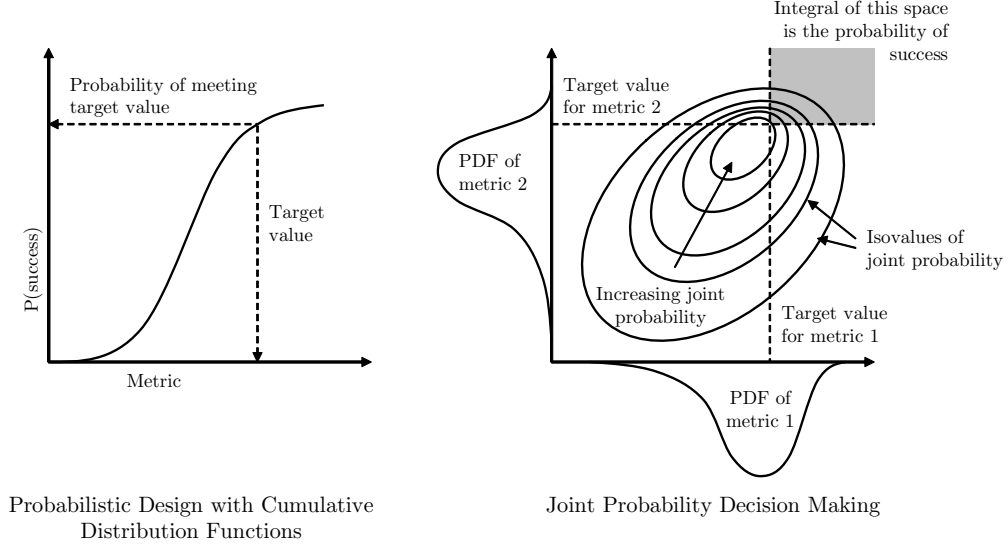
Figure 5: Probabilistic Decision Making with Cumulative Distribution Functions and Joint Probability Distributions

and interaction effects amongst the design variables. This technique is ultimately of great importance to this research.

The use of CDFs or JPDM also allows for probabilistic design of uncertain systems. There is usually some degree of uncertainty associated with various constants used within an analysis, especially those related to economic or environmental factors. These "noise" variables can be assigned ranges and probabilistically explored, much like a design space exploration. In this case, the CDF or JPDM environment does not give the portion of the design space capable of meeting a design goal; rather, it enables the designer to choose values for design variables that minimize the risk associated with uncertainty in the noise variables.

Design space exploration is only one of the uses of an integrated environment and approximation methods with statistical techniques. Technology forecasting is becoming increasingly popular as new requirements and operating environments push towards more radical solutions. Often, the state-of-the-art technologies embodied within the design space exploration exercises outlined above cannot meet these radical targets without some sort of technology infusion. Therefore, a method such as Technology Integration, Evaluation, and Selection (TIES) is used to identify promising technologies in conjunction with design space exploration [Mavris and Kirby, 1999]. TIES can be used probabilistically to determine the effect of uncertainty in the effect of each technology and thus pick the most robust suite available, ultimately reducing the risk of the design program [Kirby and Mavris, 1999]. Technology forecasting can also work in the other direction. That is, if an agency has a goal, such as reducing emissions or noise by a specified amount, Technology Impact Forecasting (TIF) can be implemented to identify the improvement required in individual technical metrics to meet this goal [Kirby and Mavris, 2002].

An area of recent interest within design space exploration methods is that of the role of

requirements in systems design. Often, the methods outlined above refer to static requirements. The same statistical techniques can be applied to systems with varying requirements, and have been demonstrated with some success in conjunction with technology integration [Mavris and DeLaurentis, 2000; Baker and Mavris, 2001; Baker, 2002]. The research conducted within this document extends the work in the field.

2.3.2   Evolving Techniques

Of final interest in modern sizing methods is the recent introduction of volume-based techniques. Before, the vehicle scaling laws considered were simply based on gross weight. Now, an increased emphasis is being placed on how the system scales volumetrically. Some preliminary "volume-balance" methods have been proposed by Raymer [Raymer, 2001]. Volume-based methods will continue to evolve as more radical, low fuel density concepts, such as hydrogen-powered low-emission vehicles, are considered [Guynn and Olson, 2002].

2.4   Multi-Mission Approaches

From its inception, sizing an aircraft required an enumeration of mission requirements. This "design mission" thus formed the basis of the fuel-required curve in the range (or endurance) related vehicle scaling laws. The design mission is met when the fuel required to propel the vehicle along the specified mission profile matches the fuel available within the vehicle. Over time, the design mission began to resemble less of what mission the vehicle would actually fly, but rather became a series of limiting conditions for what the vehicle could be expected to accomplish over a series of similar missions. As identified in the previous chapter, aircraft are increasingly being tasked with a wider variety of missions. This can make specification of a design mission difficult. One can envision several ways of incorporating disparate mission profiles into a single sizing mission, but each will likely fall under one of two main headings: size to the most critical mission, or size to a composite mission.

Sizing to the most critical mission is perhaps the simplest of these techniques and most similar to current practices. In this method, the designer first defines the necessary mission profiles and associated mission-specific performance, payload, and cost constraints. The vehicle is sized to each of these secondary missions subject to the various point-performance constraints associated to that particular mission, as well as the general non-mission specific constraints. Ultimately, the mission resulting in the largest gross weight vehicle becomes the only mission capable of meeting all of the other mission (fuel) requirements, and therefore becomes the de facto design mission. In this way, the vehicle is "overdesigned" with respect to some mission performance (range and endurance) metrics, though often others will suffer, such as off-design performance on the secondary missions and program cost.

One way to eliminate the potential for overdesign is to build a composite mission profile capable of parametrically modeling each of the specified missions. This composite mission would contain variable-length segments where necessary. These variables form a "mission space" that can be evaluated separately or together with the design space of the vehicle [Baker and Mavris, 2001]. Variation of both design and mission space values gives the designer a tool to further envision tradeoffs amongst the design variables and requirements. Ultimately, this can be quite helpful in choosing the final design mission when sizing the vehicle, and is the first step in eliminating the concept of a design mission entirely. Each point within the mission space represents a unique design mission to which the vehicle is

sized. Secondary performance variables can be tracked and traded off in such a way that a solution may not meet the most stringent mission requirements, but may be able to more affordably meet most of them.

2.4.1 Shortfalls

By its vary nature, multi-mission systems design fits naturally into the realm of Multiple Criteria Decision Making (MCDM) techniques, including such aspects as the relative importance of each mission. This frame of mind is largely ignored through sizing-based techniques listed above as they always boil down to specification of a single design mission meant to encompass all of the vehicle requirements. As system concepts become more radical and the specific missions required of the vehicle become more diverse, the chance of meeting all mission requirements (as in the first multi-mission sizing method) becomes improbable. Furthermore, "sizing" generally implies a fuel balance, which requires iteration and convergence to solve correctly. In a composite mission approach, large diversity in individual missions may entail composite mission parameter ranges that lead to numerical instability.

A final point worth mentioning is that neither of these techniques is fully adaptable to design with uncertain requirements. Certainly, the mission space model can capture uncertainty in mission-related elements, but other constraints, such as point-performance and cost may pose problems. The constraint lines can be moved, but constrained optimization techniques are not necessarily the best suited for probabilistic constraint evaluation. Instead, an unconstrained or penalty-function approach may be more appropriate for capturing uncertainty in system-level requirements.

2.4.2 Requirements Fitting for Multi-Mission Design

One problem with any form of multi-mission "sizing" is that a fuel balance is always implied. In order for a fuel balance to have any meaning, a unique design mission must be specified that may or may not accurately represent the overall mission expectations of the aircraft. As mentioned above, the fuel balance was originally created as a method to estimate gross weight for mostly single-mission aircraft based on statistical scaling laws. It is possible to run this process in reverse; that is, specify a gross weight and other pertinent design parameters and attempt to find what sorts of mission profiles the vehicle can fly. This is especially well-suited for multi-mission vehicle design because it becomes a simple matter to track pertinent individual mission performance parameters (such as segment range, endurance, and point performance). Instead of a fuel balance, a parametric sweep of gross weight and other design parameters can be made, and mission performance tracked in each case. The best multi-mission aircraft would then be the vehicle with the design parameters and gross weight that best "fit" the individual mission requirements.

This approach is attractive because it makes MCDM an integral part of the design process. Now it is possible to directly tradeoff mission performance metrics for all of the individual missions as well as other, non-mission specific metrics. This approach also mimics the "true" behavior of the system since the designer is virtually creating and test-flying a series of vehicles that make up a decision-making environment. Finally, it is well-suited for automation as most aircraft analysis codes work much better and faster without the iteration and convergence required for a fuel balance.

CHAPTER III

DESIGN AND DECISION MAKING

Almost every aspect of life involves some form of decision making, whether it is large choice such as what job to pursue or a small choice such as what type of shirt to wear. Large or small, each decision has consequences that require us to forecast the effect of various alternatives. Once the choice is made, it is a matter of time to find whether the forecast was spot-on or far off due to some unexpected event.

Engineering design also involves decision making and forecasting, though the methods used appear to be far more concrete, at least on the surface. However, when one digs down further, the number of assumptions within an analysis that is the basis of a design decision may be inaccurate. Thus, engineering design will have some degree of uncertainty.

Many design decisions involve the use of mathematical or numerical optimization to reach an "ideal" setting of variables with respect to a single criterion. Unfortunately, optimization is not a substitute for decision making. Zeleny [Zeleny, 1982] challenges his readers with the statement:

> No decision making occurs unless at least *two* criteria are present. If only one criterion exists, mere measurement and search suffice for making a choice.

This implies that single-objective optimization is not decision making at all. An analogy for the existence of multiple criteria in decision making can be made in the choice of shirt to wear. One may consider comfort, appearance, setting, and many other criteria even in this simple case. If only comfort were important than one would simply have to search through their closet and find the most comfortable shirt. Thus, optimization is more about developing and executing the right search algorithm for a single-objective problem whereas decision making is concerned with making a choice from multiple objectives.

Another necessary condition for decision-making is the availability of at least two distinct alternatives. Obviously, choosing which shirt to wear is a moot point if an individual only owns one shirt or 100 identical shirts. Of course, if the option exists to not wear a shirt at all, then a decision can be made amongst differentiated alternatives.

Unfortunately, humans are poor at making rational, repeatable decisions despite the fact that they experience decision making on a daily basis. This is especially true when there are many more than two criteria to consider. The subjective nature of decision making, along with inherent biases and lack of proper processing of information, can lead to poor judgement. Shepard [Shepard, 1964] notes:

> At the level of the perceptual analysis of raw sensory inputs, man evinces a remarkable ability to integrate the responses of a vast number of receptive elements according to exceedingly complex nonlinear rules. Yet once the profusion and welter of this raw input has been thus reduced to a set of usefully invariant conceptual objects, properties, and attributes, there is little evidence that they can in turn be juggled and recombined with anything like this facility.

Shepard continues in this work by referring to a two-dimensional experiment with linearly correlated attributes. The subjects surveyed would make a plethora of choices related

to individual biases in one or the other dimensions. He concludes that while humans are capable of making subjective decisions, one should have some sort of computational aid to the process if at all possible.

Certainly, one needs some sort of analytical means to help evaluate and select concepts with multiple attributes. What follows is a description of some important components and a few selected techniques in Multiple Criteria Decision Making (MCDM).

3.1 Pareto Optimality

An attractive concept in MCDM is the reduction in number of alternatives one evaluates. One such reduction concept is to only choose from concepts that are Pareto optimal. A Pareto-optimal solution, also known as an efficient or nondominated solution, is an alternative that is not dominated in at least one criterion. That is, there is no other feasible alternative with the same or better performance considering all criteria [Zeleny, 1982]. The locus of these points is known as a Pareto frontier (also efficient frontier, efficient surface, nondominated surface, and other permutations). This moniker refers to the work on economic theory by the Italian economist Vilfredo Pareto with regards to economic efficiency [Pareto, 1971]. A schematic of a two-dimensional "larger-is-better" Pareto frontier is given in Figure 6.
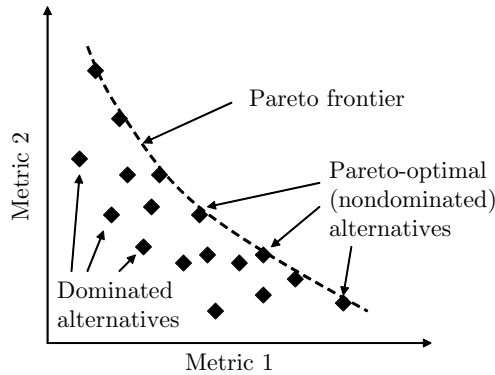
Figure 6: Two-Dimensional Pareto Frontier

In the strictest sense, one *always* wants to choose a solution from the Pareto frontier. Choosing another feasible solution not on the frontier implies that the decision maker is giving up some "free" performance in at least one of the decision metrics. In practice, however, this may not necessarily be the case. In order for Pareto optimality to hold, one must ensure that *all* of the decision metrics of interest are present, no matter what the nature of the data: cardinal, ordinal, and otherwise. Often, a decision maker may choose an alternative not on the Pareto frontier simply because another decision metric may be hard to quantify mathematically or may not be present in the data set. The remedy for this is to attempt to capture all of the decision criteria no matter what the nature of the data. In cases where subjective criteria emerge one can at least attempt to quantify the ranking relationship of the alternatives.

3.2 Axiomatic Design

While not explicitly under the heading of MCDM, Axiomatic Design is a process that seeks to resolve some of the issues with designing to multiple requirements. Notably, it attempts to remove conflict by maintaining independence amongst requirements. Suh defines two design axioms [Suh, 1990] as:

> *The Independence Axiom*: Maintain the independence of the Functional Requirements (FRs).
>
> *The Information Axiom*: Minimize the information content of the design.

The independence axiom ultimately establishes that the Functional Requirements must be independent such that only *one* is modified for any perturbation of a Design Parameter (DP). The information axiom states that the best design is the one which minimizes the number of FRs and DPs required to define the design.

An oft-used axiomatic design example is the choice of water faucet for household use as seen in Figure 7. In this case, the FRs are to control water temperature and water flow rate. A typical, "bad" design for a faucet has two handles and one spigot; one handle controls the flow of hot water and one of cold water. Here, one cannot independently control either temperature or flow rate; rather, the user must modify inputs to both handles to change one of the outputs. The Axiomatic Design approach would select a more modern faucet with a two-degree of freedom handle. In the latter case, the vertical rotation of the handle controls the flow rate and the horizontal rotation controls the temperature. Here, the independence of the FRs is maintained with respect to the DPs.

Standard design   Axiomatic design
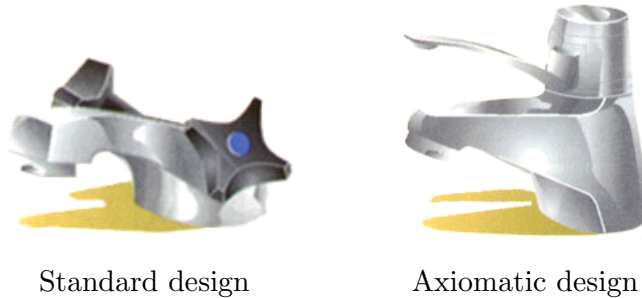
Figure 7: Axiomatic Faucet Design [MIT Axiomatic Design Group, 2005]

In theory, Axiomatic Design seems to be a good approach for design. In practice it may be much harder to enforce, especially in systems engineering situations where the design parameters are high-level abstractions of individual systems and the functional requirements are industry or government standards. Often it becomes impossible or improbable to discern individual inputs for the design parameters. An aggregate set of requirements may be possible but difficult to relate. If a decomposition-based approach were used, it may become possible to create a set of independent functional requirements with transparency to the original requirements, though it would be much harder to enforce independence of design parameters. The attractive feature of independent requirements lies in the decision-making itself: if the requirements are truly independent there should be no bias from one solution to the next.

3.3 Multiple Criteria Decision Making Techniques

Multiple Criteria Decision Making is a necessarily broad subject area. There are many classes and subgroups of methods depending on the nature of the criteria considered, the involvement of the decision maker, and the nature of the objective sought. Though the opinions of many authors differ on the subject, in this document MCDM will be used to refer to two different classes of methods: Multiple Attribute Decision Making (MADM) and Multiple Objective Decision Making (MODM) [Yoon and Hwang, 1995]. Some consider Multiple Attribute Utility Theory (MAUT) to be under this heading as well, but this subject will not be covered in detail here.

The principle difference between MADM and MODM techniques is the pertinent application. MADM techniques are focused on ranking and selection of a few options from a discrete pool of alternatives, usually not subject to constraints [Hwang and Yoon, 1981]. On the surface, MADM is more appropriate when a decision maker must choose from a pool of predefined concepts, such as the government's choice of five competing fighter configurations from different companies or one's choice of a stock portfolio. MODM techniques are more appropriate for design applications, as they focus on multiple objectives within a continuous space and are subject to active constraints [Hwang and Masud, 1979]. Both techniques capitalize on various weighting techniques to determine ranking or preference relationships between the various criteria.

A variety of techniques are available for both MADM and MODM depending on the nature of the information available to the decision maker or algorithm. Figures 8 and 9 are reproduced from the work of Hwang and his colleagues [Hwang and Yoon, 1981; Hwang and Masud, 1979] and display a taxonomy of MADM and MODM techniques, respectively. Note that MADM techniques vary by the type of information available whereas MODM techniques are categorized by both the type of information and the stage at which this information is needed.

The decisions made within aerospace systems design usually refer to cardinal data; that is, data associated with a quantity but not necessarily a preferential order (note that a ranking relationship may be derived from cardinal data if a goal or direction of improvement is noted). The data retrieved from engineering analyses are, with few exceptions, cardinal. Examples include gross weight, cost, sustained turn rate, and takeoff field length, to name a few. Some more subjective criteria may be ordinal in nature, that is, simply ranked with respect to the other alternatives. An example may be a qualitative measure of reliability as low, medium, or high. This is far less common in engineering analysis. Therefore, only some of the cardinal methods shown in Figures 8 and 9 or their associated outgrowths will be elaborated upon further. For a more complete description of these techniques the reader is directed to more comprehensive references [Hwang and Masud, 1979; Hwang and Yoon, 1981; Zeleny, 1982; Triantaphyllou, 2000].

3.3.1 The Ideal Solution

A powerful concept in cardinal MCDM techniques is that of the *ideal solution*. This is the solution that embodies the best answer within the design domain for each of the attributes. Mathematically, the ideal solution can be stated as

$$Y^* = (y_1^*, \, y_2^*, \, \ldots, \, y_n^*) \tag{1}$$

where Y^* is the ideal solution and y_n^* is the best value of the nth attribute. This solution is often made up of attributes from multiple alternatives (if such a solution was available

Type of information from the decision maker | Salient feature of information | Major classes of methods

Multiple Attribute Decision Making

No information → Dominance, Maximin, Maximax

Information on attribute:
- Standard Level → Conjunctive method, Disjunctive method
- Ordinal → Lexicographic method, Elimination by aspects, Permutation method
- Cardinal → Linear assignment mtehod, Simple additive weight method (SAW), Hierarchical additive weighting method, ELECTRE, TOPSIS

Information on alternative:
- Marginal rate of substitution → Hierarchical tradeoffs
- Pairwise preference → LINMAP, Interactive SAW method
- Order of pairwise proximity → MDS with ideal point

Figure 8: A Taxonomy of Methods for Multiple Attribute Decision Making [Hwang and Yoon, 1981]

Stage at which information is needed | Type of information | Major classes of methods

Multiple Objective Decision Making

No articulation of preference information → Global criterion method

A priori articulation of preference information:
- Cardinal information → Utility function, Bounded objective method
- Ordinal and cardinal information → Lexicographic method, Goal programming, Goal attainment method

Progressive articulation of preference information (interactive methods):
- Explicit trade-off → Method of Geoffrion and interactive goal programming, Surrogate worth trade-off method, Method of satisfactory goals, Method of Zionts-Wallenius
- Implicit trade-off → STEM and related methods, SEMOPS and SIGMOP methods, Method of displaced ideal, GPSTEM method, Method of Steuer (interactive MOLP method)

A posteriori articulation of preference information (nondominated solutions generation method):
- Implicit trade-off → Parametric method, Epsilon-constraint method, MOLP methods, Adaptive search method

Figure 9: A Taxonomy of Methods for Multiple Objective Decision Making [Hwang and Masud, 1979]

18

with one alternative, it becomes the obvious choice). Sometimes the concept of the negative ideal solution becomes necessary for certain MADM techniques. As its name implies, the negative ideal solution has the opposite definition as the positive ideal. This represents an aggregate of the worst attributes from the available alternatives. Figure 10 shows a simple schematic of the positive and negative ideal solutions for a finite set of alternatives in two "larger is better" dimensions.
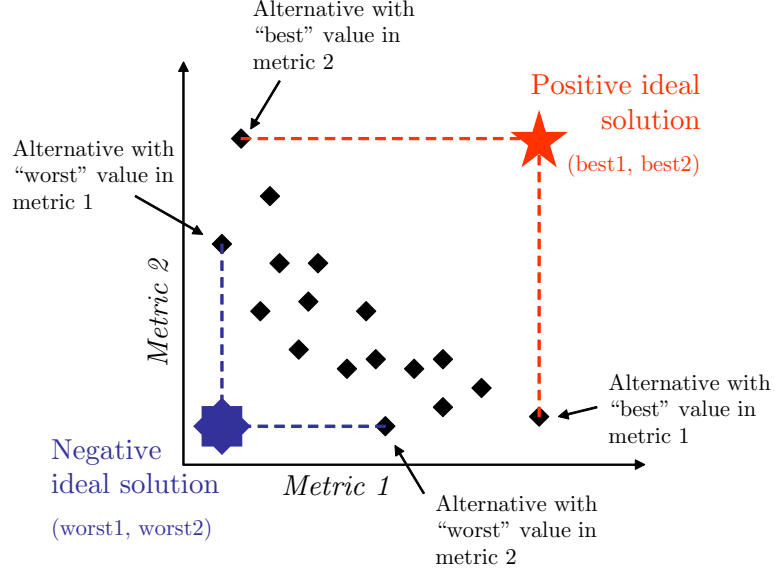
Figure 10: Positive and Negative Ideal Solutions from Several Alternatives

3.3.2 Simple Additive Weighting

Perhaps the most elementary, and most popular, cardinal MADM technique is the Simple Additive Weighting (SAW) method, also known in various forms as the Weighted Sum (WS) or Overall Evaluation Criterion (OEC) method. In these techniques, the attributes of each alternative is normalized first, usually with respect to the best attribute value amongst the pool of alternatives (i.e. the respective value in the positive ideal solution). Sometimes the values are normalized by a vector sum of the characteristics of each attribute. Next, a series of numerical weights are prescribed by the decision maker. These can be based on subjective observations by experts or on more advanced weight generation methods, to be covered later. These weights are usually normalized such that their sum is equal to one. Each alternative is evaluated by multiplying the weights times the normalized score in each respective attribute, then summing up the total. The alternative with the highest score is considered most preferable. Mathematically, SAW (when normalized by the best attribute) figures as

$$A^* = A_i \,|\, \max \frac{\sum_{j=1}^{n}(w_j \, \overline{y}_{ij})}{\sum_{j=1}^{n}(w_j)} \tag{2}$$

where A^* is the best alternative, A_i is the ith alternative, n is the number of attributes, w_j is the weight of the jth attribute, and $\overline{y}_{ij}$ is the jth normalized attribute of the ith alternative.

If "larger is better" for all y_j, then the normalization follows as

$$\overline{y}_{ij} = \frac{y_{ij}}{y_j^*} \tag{3}$$

where y_{ij} is the "raw" value of the jth attribute of the ith alternative. For a "smaller is better" case, the reciprocal of (3) is used.

Though simple and easy to understand, SAW and related methods are not well suited for rigorous decision making. Most notably, these methods will never pick a non-convex point on a Pareto frontier and will instead only choose the extremes [Mullur et al., 2003]. This is best visualized by projection of a non-convex frontier onto a series of SAW indifference curves. An indifference curve is a line along which a method will receive the same score. Figure 11 shows this for a two-dimensional "larger is better" series of alternatives with a non-convex Pareto frontier. Here, SAW will always rank alternatives A and B higher than alternative C, even though the latter alternative may be the best available mix of performance available.
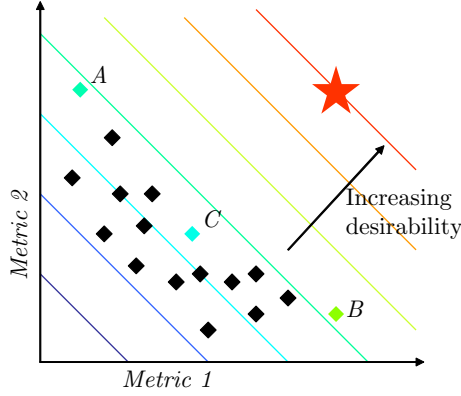
Figure 11: SAW Indifference Curves Projected Onto Non-Convex Pareto Frontier

3.3.3 TOPSIS

One MADM technique that attempts nonlinear decision-making is the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). The basic idea of TOPSIS is to rank the solutions based on the combination of Euclidean distance from the positive *and* negative ideal solutions. The concept of Euclidean distance fits well into the spatially-oriented minds of most decision makers.

Conceptually, TOPSIS does not begin much differently than the SAW-related methods. Of primary importance is the normalization of the individual attributes, though TOPSIS accomplishes this through the 2-norm via

$$\overline{y}_{ij} = \frac{y_{ij}}{\sqrt{\sum_{i=1}^{m} y_{ij}^2}} \tag{4}$$

where m refers to the number of alternatives and the other notation is as in (3). Once normalized, the separation from the positive and negative ideal solutions is measured for

each alternative through

$$S_i^* = \sqrt{\sum_{j=1}^{n} \left(w_j(\overline{y}_{ij} - y_j^*)\right)^2} \qquad (5)$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(w_j(\overline{y}_{ij} - y_j^-)\right)^2} \qquad (6)$$

where S_i refers to the separation measure, the subscript '$*$' refers to the positive ideal solution, and '$-$' refers to the negative ideal solution. All other notation is as before.

These separation measures are combined into a single measure dubbed the *closeness to the ideal solution* for each alternative. This is figured from

$$C_i^* = \frac{S_i^-}{S_i^* + S_i^-} \qquad (7)$$

where C_i^* is the closeness measure for the ith alternative. The solutions are then ranked in descending order; i.e. the alternative with the highest value for C_i^* is considered the "best," with the next-highest value as the "second-best," and so on.

The 2-norm would seem to indicate that this method is capable of capturing some non-convex Pareto optimal solutions. However, the use of both the positive and negative ideal solution measures flattens the indifference curves halfway between the positive and negative ideals. Therefore, this still leads to poor resolution and ranking of solutions on non-convex Pareto frontiers. Figure 12 shows two cases of TOPSIS, both for "larger is better" cases in two dimensions.
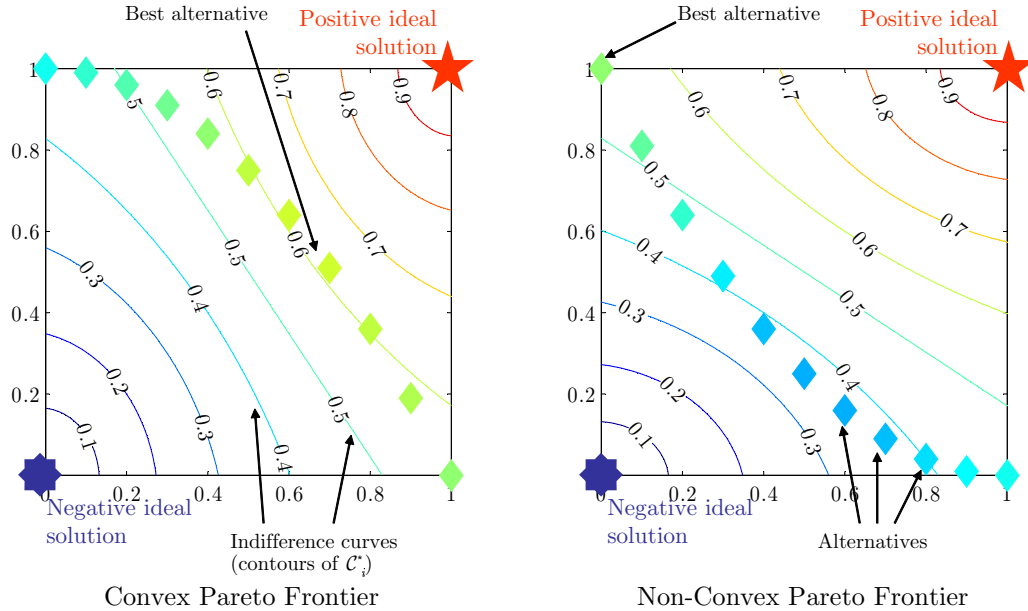
Figure 12: TOPSIS Solutions for Two Non-Convex Pareto Frontiers

Figure 12 shows that TOPSIS does indeed have different indifference curves than SAW but is still not capable of ranking non-convex alternatives and instead will rank one of the

extreme alternatives in this situation. The behavior of TOPSIS indifference curves is cited by Yoon [Yoon and Hwang, 1995] as being rational. He states:

> When a [decision maker] recognize's ones solution is closer to the negative-ideal than to the positive-ideal, the [decision maker] is inclined to pick an alternative that consists of the best and worst attributes rather than one with two worse attributes. For example, one might want to get one A grade and one F grade rather than two D grades.

This choice reflects one of the caveats of TOPSIS: the assumption of monotonically increasing utility. Unfortunately, this may not always be a valid assumption. Therefore, methods that are able to capture some shallow non-convex solutions may be more appropriate when a decision maker is willing to enable larger tradeoffs to achieve a compromise.
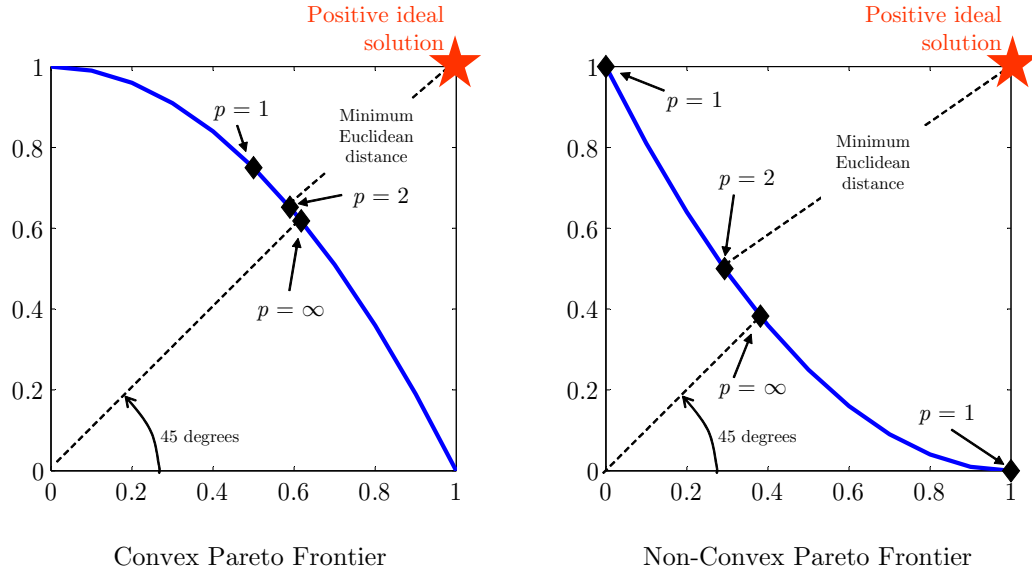
3.3.4 Compromise Programming

Compromise Programming (CP) is a quasi-MODM technique that is a combination of Multi-objective Linear Programming and Goal Programming [Zeleny, 1982]. The former approach is an extension of the popular linear programming techniques, such as the Simplex method [Chvátal, 1983] for problems with multiple objectives, hence its MODM identification. Goal programming is usually a linear programming technique as well, except with modified objective functions to reflect a specific goal. This is more of a "nominal the best" style of optimization where the goals frequently lie within the design domain but may not necessarily be a maximum or minimum of the alternatives within the concept space (hence the quasi-MODM identification). Compromise programming is very flexible and can be used for nonlinear problems. It too uses the concept of the positive ideal solution and in some cases the negative ideal solution, sometimes referred to in the literature as the "anti-ideal."

A popular method for CP uses the positive and negative ideal solutions (or, in a continuous space, the "best" and "worst" values) and has an objective function of the form [Zeleny, 1982; Vanderplaats, 1999]

$$F(\vec{x}) = \left\{ \sum_{j=1}^{n} \left[\frac{w_j(F_j(\vec{x}) - F_j^*)}{F_j^- - F_j^*} \right]^p \right\}^{\frac{1}{p}} \tag{8}$$

where $F(\vec{x})$ is the overall objective function to be minimized, $\vec{x}$ is a vector of the design variables, $F_j(\vec{x})$ is the jth objective (attribute) function, F_j^* is the "best" value of $F_j(\vec{x})$ (analogous to y_j^* in (3)), F_j^- is the "worst" value of $F_j(\vec{x})$, and p is a parameter dependent on the type of norm to be used in the optimization. The most common values for p are 1, 2, and ∞. When $p = 1$ the CP algorithm is essentially a weighted sum technique. For $p = 2$ CP finds the solution the minimum Euclidean distance away from the positive ideal, and for $p = \infty$ CP will find a solution that minimizes the maximum deviation from the ideal. The changes in solution for different values of p is demonstrated for a two-dimensional "larger is better" convex and non-convex case in Figure 13.

Compromise programming seems well-suited for design problems because it is essentially a MODM (hence continuous) technique but allows the user to specify goals. It also allows the decision-maker to tailor preferences in solution method via p. Furthermore, it is adaptable to a variety of normalization methods in addition to the ideal solution method in (8).

Figure 13: Compromise Programming Solutions for Different Values of p

3.3.5 MCDM for Systems Design

Systems design is not necessarily well-suited to a traditional design and decision making environment. Most detail design environments require fixed requirements and constraints, one or very few objectives, and a continuous or mixed continuous-discrete design space. The solution is often based on closed-form analysis. In contrast, most decision making environments are set up to handle a discrete solution space, have several attributes to consider for each alternative, and are reliant on highly subjective methods. The former is well-suited to detail design optimization and the latter is better for making decisions from a pool of predefined concepts and "what-if" scenarios such as risk assessment. Effective systems design is often a combination of both environments. Some design and MDO frameworks were mentioned in Chapter II; these, combined with the MCDM techniques outlined in this chapter, form the building blocks for effective multiple criteria systems design.

3.4   Case Study: Notional Multi-Role Fighter

A case study was performed on the requirements selection and design of notional multi-role fighter in an attempt to gain more information on the pertinent issues associated with large-scale systems design problems. This fighter was based on the requirements growth of the McDonnell-Douglas F/A-18C Hornet configuration to the F/A-18E Super Hornet series of aircraft proposed by Boeing. The Super Hornet is supposed to supplement and eventually replace the missions currently performed by three aircraft in the U.S. Navy's inventory [Young et al., 1998]: the F/A-18C Hornet, A-6F Intruder, and F-14D Tomcat. The general idea of this study was to attempt to "grow" an existing aircraft configuration to attempt to meet or exceed the mission capabilities of the legacy aircraft. The study was conducted in such a way as to keep the growth configuration flexible to eventually investigate if the F/A-18C was the most effective configuration to work from. The study that follows is a summary from some recently published literature on the subject [Borer and Mavris, 2003,

2004]. The reader is referred to these documents for a more comprehensive review.

3.4.1 Problem Formulation

The main formulation of this study involved six steps: Requirements Classification, Baseline Concept Definition, Creation of Integrated Environment, Decision Space Population, Exploration of Requirements Tradeoffs, and finally, Decision Making. The first step was universal for all configurations, while steps two through five could be repeated for different concepts if necessary. The final decision could be based on the results from several concepts. This study only considered the F/A-18C configuration but the overall formulation was generic enough that any (or several) baselines could be used.

   The requirements classification followed from examination of the missions of the legacy aircraft the Super Hornet was meant to replace. However, some of the latest data on the legacy aircraft was not available, so data was used from earlier models of the same aircraft. The mission requirements and profiles were collated from the Standard Aircraft Characteristics (SAC) charts of the F/A-18C [Naval Air Systems Command, 1996], A-6E [Naval Air Systems Command, 1984], and F-14A [Naval Air Systems Command, 1976]. In total, these aircraft performed 21 missions, though there was redundant capability amongst the legacy aircraft so they only accounted for nine distinct mission profiles (albeit with different individual capabilities). For the most part, the most critical (in terms of payload-range) mission parameters were chosen when multiple airframes flew the same mission profile. These missions are depicted in Table 1 with the "dominant" configuration in bold.

Table 1: Notional Multi-Role Fighter Missions

Mission	Aircraft
Hi-Hi-Hi	F/A-18C
	F-14A
	A-6E
Lo-Lo-Lo	F/A-18C
	F-14A
	A-6E
Hi-Lo-Hi	F/A-18C
	A-6E
Hi-Lo-Lo-Hi / Interdiction	F/A-18C
	F-14A
	A-6E
Lo-Lo-Hi	**F-14A**
Close Support	F/A-18C
	F-14A
	A-6E
Fighter Escort	F/A-18C
	F-14A
Deck-Launched Intercept	**F/A-18C**
	F-14A
Combat Air Patrol	**F/A-18C**
	F-14A

The analysis environment utilized the requirements fitting formulation for multi-mission design outlined in section 2.4.2. This served as a test for the requirements fitting technique and ensured that the analysis codes would run smoothly for all nine missions. A total of 44 decision metrics were tracked: four mission performance metrics for each of the nine missions, five individual mission performance metrics, and three non-mission specific metrics. These are listed in Table 2. Note that some of these metrics were inputs into the analysis code. Also, no explicit cost metric was available so gross weight was used as an estimate of life-cycle cost. Though perhaps a poor estimate, this exercise was meant to be exploratory in nature and therefore appropriate enough.

**Table 2:** Notional Multi-Role Fighter Decision Metrics

Description	Units	Total #
Maximum Mach number	-	9
Maximum sustained load factor	g's	9
Specific excess power	ft / sec	9
Total range	nmi	9
Combat Air Patrol acceleration Mach number	-	1
Combat Air Patrol time on station	min	1
Close Air Support time on station	min	1
Deck-Launched Intercept dash Mach number	-	1
Hi-Lo-Lo-Hi / Interdiction low-level dash distance	nmi	1
Approach speed	kts	1
Approach single engine rate of climb	ft / sec	1
Takeoff gross weight	lbs	1

The core of the analysis was NASA Langley's FLOPS [McCullers, 1984, 1998]. This code is capable of fuel-balance sizing or direct mission analysis of a configuration for specified gross weight, so it was used in the latter mode in keeping with the ideas of the requirements fitting process. A baseline input file for the F/A-18C configuration was calibrated to actual performance and aerodynamic data [Naval Air Systems Command, 1996; McDonnell-Douglas Corporation, 1996]. A series of spreadsheets were used to scale the input FLOPS file for the various system-level inputs and to figure changes in store drags for the various missions. These files were wrapped in a ModelCenter [Phoenix Integration, 2005] environment for ease of execution and parsing data.

This integrated environment was executed multiple times for various settings within a Design of Experiments (DoE) table for creation of quadratic Response Surface Equations (RSEs). The Design of Experiments and subsequent creation of the RSEs was handled with JMP [SAS Institute, 2005], a statistical analysis software package. Once created, the RSEs were validated through a series of statistical tests within JMP and finally through a comparison of several random input cases. All tests indicated that the RSEs were good

approximations to the data provided by the original analysis environment.

The RSEs served as a rapid way to evaluate the various responses in a probabilistic MCDM environment. The overall scheme involved creation of a random population of 100,000 data points. The RSEs were used to rapidly create a 44-element vector of attributes for each of these 100,000 alternatives. Note that no screening was performed on these alternatives to determine which, if any, were not Pareto optimal. This was due to resolution issues associated with generating a Pareto frontier in very large problems (this will be discussed in general later). TOPSIS was used to rank each of the alternatives subject to a complex scheme for determination of weighting factors. Ultimately, each of the 44 metrics had its own importance weight factor. The uncertainty in the actual requirements was simulated via uncertainty in the weight factors. As such, 5,000 quasi-random weighting scenarios were developed in a Monte Carlo Simulation for TOPSIS evaluation; hence, each of the 100,000, 44-metric alternatives were ranked 5,000 times via TOPSIS. In this fashion, the idea was not to pick the design that was ranked first in any individual TOPSIS execution, but rather the design that most often appeared in the top few percent of every TOPSIS trial. This design would be the most robust with respect to unexpected changes (uncertainty) in the requirements. Figure 14 illustrates the prominent features of this analysis.
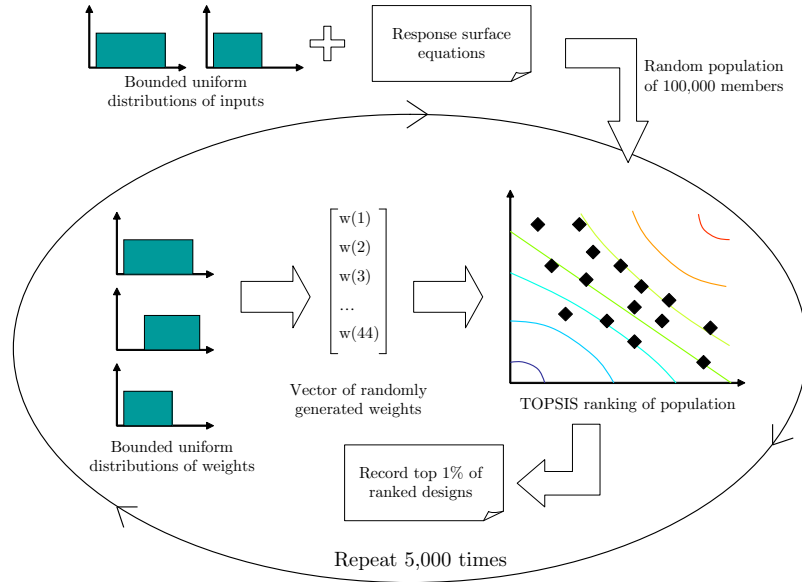
Figure 14: Notional Multi-Role Fighter Problem Formulation and Execution

3.4.2 Results and Implications

The top one percent of the TOPSIS-ranked population were recorded for each of the 5,000 MCS trials. Of these 5,000 trials, one particular alternative appeared in the top one percent of the population 2,365 times, and 26 alternatives appeared over 2,000 times. If the top designs all have similar inputs and outputs, then any one of these designs represents one that will be invariant with small changes in preference.

Unfortunately, this was not the case. The resulting input and output histograms for the top recurring designs showed considerable spread, including gross weight. Furthermore, the

recurrence was much smaller than hoped. At best, the top recurring design was in the top one percent 2,365 out of 5,000 trials, slightly more than 47% of the time. This level of risk is likely unacceptable for most programs. This could perhaps be increased with a smaller range and refined distributions within the MCS trials, but seemed to be indicative of bigger problems.

One such problem with this formulation was the incredible number of requirements. Unfortunately many of the requirements were highly dependent on the same parameters as others, potentially creating "double-weighting" (and more) situations within the TOPSIS trial. That is, if two requirements were essentially the same metric, then including them in any MCDM approach will act as if that one metric were twice as important.

One final problem with any MCDM formulation is a lack of threshold values. Often, there is a threshold beyond which a decision maker becomes indifferent to the value of a particular metric. TOPSIS and other MCDM formulations will essentially try to exploit the responses with the greatest slope in finding a solution (this slope may be partially mitigated with the weighting factors). However, this may not be desirable once a metric reaches or approaches a certain value, and instead the optimization effort should be spent in other dimensions even if they have a lower slope.

3.4.3   Research Directions

In all, the notional multi-role fighter case study was a success. Though the results themselves were inconclusive, the exercise demonstrated some of the issues associated with bringing MCDM to large-scale systems design problems.

Systems design involves decisions made early on in a process considering such things as risk, performance, and cost. As mentioned before, aircraft design is often fraught with uncertainty, especially with respect to requirements and constraints. As such, an effective multiple criteria systems design environment would be one that could handle uncertainty in requirements specification. It would also need to be capable of identifying key tradeoffs and the requirements which drive these tradeoffs.

As a systems design problem grows to encompass more requirements, the complexity associated with its MCDM environment grows. More solutions become Pareto optimal, and the addition of uncertainty further muddies the water when attempting to find the best compromise solution.

Four principal questions arise from the observations made thus far. These have to do with the idiosyncracies associated with probabilistic design and MCDM for many requirements. These research questions are identified as follows:

(R1) Is there a design formulation that is better suited to integration of probabilistic MCDM techniques?

(R2) What is the best solution to MCDM with uncertain requirements?

(R3) What are the key tradeoffs amongst a large set of requirements?

(R4) Is it possible to identify an ideal tradeoff?

The reader is asked to keep these questions in mind as the research plan unfolds throughout the rest of this document. The first two questions relate to some of the probabilistic design methods already identified, but the latter two invite a new formulation for MCDM to help determine relative importance of metrics and the selection of tradeoff values.

CHAPTER IV

DECISION MAKING FOR LARGE-SCALE PROBLEMS

The techniques described in Chapter III begin to give an appreciation for the choices available when choosing a formulation for design and decision making. The MCDM methods especially have a unique taxonomy depending on the nature of the decision data: discrete, continuous, ordinal, cardinal, and more. Unfortunately, all decision making methods have some prominent drawbacks, especially with respect to large-scale problems. Some of these issues were discussed in the notional multi-role fighter case study presented in the previous chapter. The most notable drawbacks are the assumption of monotonically increasing utility for each metric and also the dependence of the decision metrics on others. However, these issues are not insurmountable and can be dealt with within the appropriate framework. What follows is the development of a probabilistic MCDM model that is better suited to large-scale problems. This formulation is developed and demonstrated in parallel with an example problem to further illustrate the principles at work.

4.1 *Generalized Probabilistic MCDM Formulation*

Fundamental to the description of a large-scale MCDM environment is the creation of a generalized probabilistic MCDM formulation. This environment should be able to capture uncertainty in the requirements themselves or in the decision maker's preference regarding one metric over another. Three salient characteristics of such a formulation are as follows:

- Probabilistic

- Lack of internal constraints;

- Implicit tradeoffs.

As with the notional multi-role fighter, the probabilistic portion of this formulation can be handled through the relative weights in the MCDM environment. A probabilistic relative weight is analogous to the decision maker's uncertainty regarding the relative importance of one metric to the next. It follows that the consistently higher-ranked solution is robust (relatively invariant) with respect to the decision metrics and therefore represents the best probabilistic solution.

Lack of internal constraints is very important in such a formulation because these can bias an integrated environment or cause bad fits to data if surrogate models are used (as is often necessary for probabilistic analyses; refer to Chapter II for more details). These internal constraints may require iteration and convergence that are not transparent to the designer and therefore can cause problems when specifying other requirements. Furthermore, many constraints may actually be requirements or decision metrics and therefore it is important to allow the decision maker external access to these values.

Finally, any non-trivial decision making formulation requires tradeoffs. For large problems, the sheer number of tradeoffs is a combinatorial of the number of metrics, i.e. four requirements involves up to $3 + 2 + 1 = 6$ tradeoffs, ten requirements can involve

$9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 45$ tradeoffs, and so on. Therefore, it quickly becomes difficult to handle explicit tradeoffs. These can be made implicit with many MCDM formulations simply through definition of a positive and negative ideal solutions along with relative weights. Compromise programming is an attractive option because it can normalize the data to the positive and negative ideal solutions, is a continuous formulation, and allows the decision maker flexibility regarding the type of tradeoff to be made via the 1-, 2-, or ∞-norm. Compromise programming will also *always* capture a nondominated solution [Ismail-Yahaya and Messac, 2002].

The generalized probabilistic MCDM formulation can be formally summarized via the statement of three hypotheses:

(H1) Effective Multiple Criteria Decision Making requires an open, internally unconstrained integrated environment.

(H2) Requirements uncertainty can be captured through probabilistic relative importance weights of the individual metrics. It follows that the concept that is most robust with respect to change in the requirements is the concept whose ranking is most invariant with respect to changes in these weights.

(H3) Tradeoffs amongst multiple requirements can be implicitly handled through a decision-making algorithm capable of compromise. This compromise is best characterized with respect to the positive and negative ideal solutions.

4.1.1 Generating Alternatives

All MCDM techniques require a dataset from which to make decisions. This dataset is generally discrete for MADM techniques and continuous in MODM, though this is not a rule. It is possible to represent a continuous space with a collection of random discrete points and just as possible to represent most discrete points with some form of continuous function if necessary. In this vein, "generating alternatives" need not necessarily mean the generation of discrete alternatives. Rather, it may just as likely refer to characterization of the overall continuous decision space for later MCDM evaluation.

Ideally, a decision maker wishes to choose from those solutions that are Pareto optimal. Choosing from this subgroup ensures that all tradeoffs, implicit or otherwise, would occur amongst nondominated alternatives. While an attractive option in theory, resolution of the Pareto frontier becomes difficult as the number of objectives increases. This has been documented for multiobjective problems where evolutionary algorithms are used to glean information on the Pareto frontier [Deb, 2001]. Here, an initial population is necessary and an estimate can be made as to what fraction of the population size will be nondominated. Guidance for population size selection for up to 10 objectives can be found in Figure 15.

Furthermore, the computational effort required to resolve nondominated solutions increases nonlinearly as the population size increases. As can be inferred from Figure 15, even large populations contain mostly nondominated solutions. This seems to imply that pre-screening dominated solutions is not worth the computational savings realized from a reduction in number of decision alternatives for problems with many objectives. The analogy of fraction of nondominated population size can be carried over to continuous decision spaces. Here, the Pareto optimal solutions would be a continuous subset of the decision space. As the number of decision metrics (objectives) increases, the portion of the decision space that is completely dominated decreases until it is insignificant. This leads to the next hypothesis regarding large-scale MCDM:
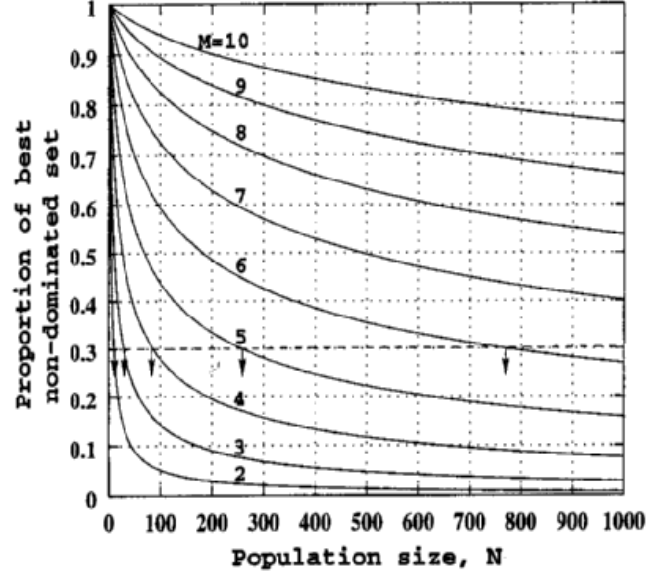
Figure 15: Nondominated Population Fraction for Increasing Number of Objectives [Deb, 2001]

(H4) Resolution of the Pareto frontier becomes inconsequential as the number of decision metrics increase. For a large number of metrics, the complete design space is an adequate representation of the subset of nondominated solutions.

4.2   MCDM Example Problem: Beam Design

An example problem will be used throughout the rest of this chapter to further illustrate some of the issues and new approaches for large-scale MCDM. This example is a beam design problem with requirements calculated from simple structural analysis principles. These closed-form requirements are easily understood and should resonate with most engineers.

The design problem is a three-dimensional vertical beam with a fixed end condition. The beam has two design variables, both relating to cross-section. These are width (w) and aspect ratio (AR), the latter of which is defined as the ratio of width to depth. The beam length and material properties are constants. Figure 16 gives the pertinent information for this example problem.

Its requirements are characterized in three different loading conditions: one end-loaded compressive case and two cantilever cases. Each of these loading conditions has specific requirements for displacement and factor of safety for ultimate failure (maximum stress). The compressive case has an Euler buckling factor of safety requirement. Finally, the beam is to be designed to minimize weight. Figure 17 illustrates the beam loading conditions.

Inspection of Figure 17 shows that loading conditions two and three are redundant. The maximum stress factors of safety for these two cases will be identical and the maximum displacement of loading condition three will be dominated by loading condition two. This is deliberate; the idea is to demonstrate the issues associated with redundant requirements in cases where they may not be so apparent. To further simplify notation, these eight

Length (L) = 2.0 m

Design Variables:

Width (w) = 0.02 to 0.06 m

Aspect Ratio (AR = w/d) = 1 to 2

Material: 2014-T6 aluminum

Density (r) = 2,790 kg/m^3

Young's Modulus (E) = 7.31x10^{10} Pa

Ultimate Strength (s$_{ult}$) = 4.69x10^8 Pa
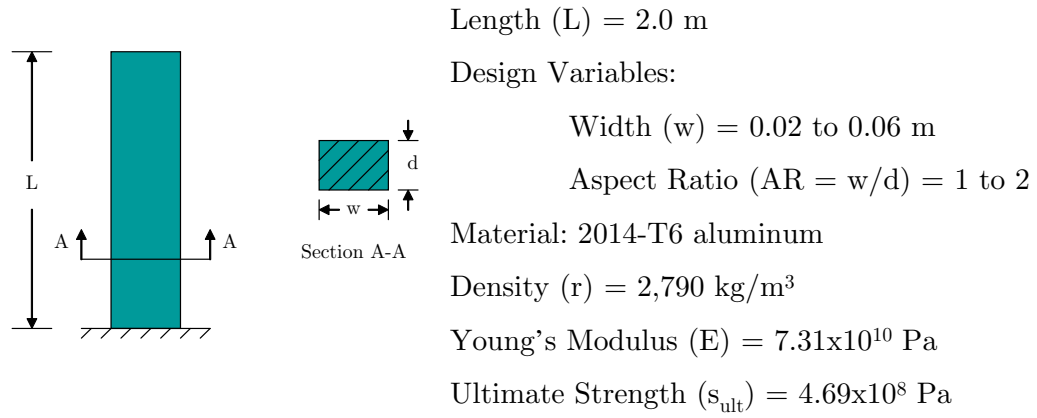
Figure 16: Pertinent Dimensions and Material Properties of Beam Design Problem

Loading Condition 1
(LC1)

P_1 = 30,000 N

Loading Condition 2
(LC2)

P_2 = 5,000 N

Loading Condition 3
(LC3)

P_3 = 10,000 N
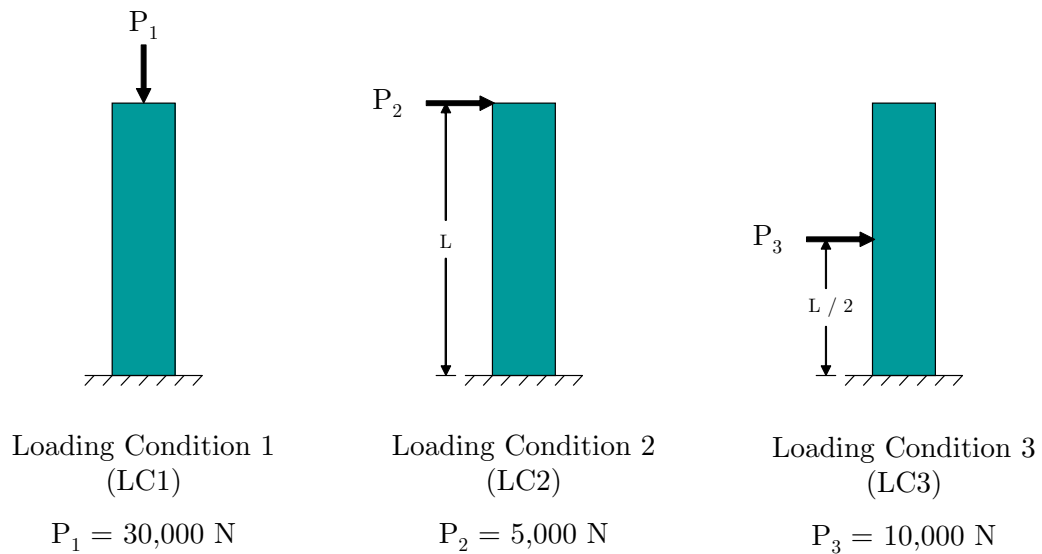
Figure 17: Loading Conditions for Beam Design Problem

requirements are labeled as follows (with LC referring to the appropriate loading condition):

- R_1: Euler buckling factor of safety (LC1)

- R_2: ultimate compressive failure factory of safety (LC2)

- R_3: ultimate bending failure factor of safety (LC2)

- R_4: maximum compressive displacement (LC1)

- R_5: maximum bending displacement (LC2)

- R_6: total beam weight (all loading conditions)

- R_7: ultimate bending failure factor of safety (LC3)

- R_8: maximum bending displacement (LC3)

The functional form of the eight requirements are found from classical structural analysis techniques [Young and Budynas, 2001; Hibbeler, 1997]. Before discussing the functional form of the equations, there are some preliminaries to consider:

$$d = \frac{w}{AR} \tag{9}$$

$$A = wd \tag{10}$$

$$I_w = \frac{1}{12}wd^3 \tag{11}$$

$$I_d = \frac{1}{12}w^3d \tag{12}$$

where d is the depth of the beam as indicated in Figure 16, A is the cross-sectional area, I_w is the second moment of area about the width axis, and I_d is the second moment of area about the depth axis. With these in mind, the functional form of the eight requirements can be stated both with respect to the notation of equations (9) through (12) and with respect to the design variables w and AR:

$$R_1 = \frac{\pi^2 E I_w}{4 P_1 L^2} \qquad f\left(\frac{w^4}{AR^3}\right) \tag{13}$$

$$R_2 = \frac{\sigma_{ult}}{\frac{P_1}{A}} \qquad f\left(\frac{w^2}{AR}\right) \tag{14}$$

$$R_3 = \frac{\sigma_{ult}}{\frac{P_2(\frac{w}{2})}{I_d}} \qquad f\left(\frac{AR}{w^3}\right) \tag{15}$$

$$R_4 = \left(\frac{P_1}{A}\right) LE \qquad f\left(\frac{AR}{w^2}\right) \tag{16}$$

$$R_5 = \frac{P_2 L^3}{3 E I_d} \qquad f\left(\frac{AR}{w^4}\right) \tag{17}$$

$$R_6 = \rho A L \qquad f\left(\frac{w^2}{AR}\right) \tag{18}$$

$$R_7 = \frac{\sigma_{ult}}{\frac{P_3(\frac{L}{2})(\frac{w}{2})}{I_d}} \qquad f\left(\frac{AR}{w^3}\right) \tag{19}$$

$$R_8 = \frac{\left(\frac{5}{8}\right) P_3 L^3}{6 E I_d} \qquad f\left(\frac{AR}{w^4}\right) \tag{20}$$

32

where E refers to the Young's modulus of elasticity, P_1 through P_3 is the load associated with loading conditions one through three, respectively, L refers to the beam length, and ρ refers to the material density.

Inspection of the functional forms given in equations (13) through (20) indicates that only five of these requirements are linearly independent. As expected, all of the LC3 requirements are the exact same functional form as the LC2 requirements; R_7 will be perfectly correlated with R_3, and R_5 will be perfectly correlated with R_8. Furthermore, two seemingly independent requirements, R_2 and R_6, are perfectly correlated. However, they are exactly opposite in terms of direction of improvement, but they will effectively cancel each other out if the assumption of monotonically increasing utility is made. Also, the presence of two functionally opposing objectives indicates that every solution within the design space is Pareto optimal since at least one of these two requirements will always be nondominated.

Basic engineering intuition leads to some assertions regarding this problem. First, the requirements of loading condition three should be completely dominated by loading condition two and therefore should not be a factor in the design problem (as long as the constraints, if any, for LC3 are the same or lower than those for LC2). Further, the apparent "design drivers" appear to be Euler buckling and bending stress or displacement. Compressive failure factor of safety and compressive deflection should be minimal, and therefore inconsequential to the final design.

4.2.1 Beam Analysis and Design Procedure

The MCDM formulation of the beam design problem emulates that of the notional multi-role fighter as well as the general procedure outlined in section 4.1. The beam analysis code was a series of files and functions created in Matlab [The Mathworks, Inc., 2005]. Though this analysis environment was very fast, this environment was used to create response surface equations of the eight requirements for a specified range of the design variables. The region of interest was for a value of w from 0.05 to 0.10 meters and a value of AR from 1.0 to 2.0. These values were varied within a five-level design of experiments to create third-order, fully interacting response surfaces for each requirement. Third-order response surfaces were used because second-order surfaces could not provide a good approximation of the original equations. These response surfaces are shown in Figure 18 for normalized inputs from -1.0 to 1.0. The answers of the response surfaces were compared randomly to determine their error with respect to the original anlayses. The actual results are plotted versus the RSE results in Figure 19.

Unless otherwise specified, the surrogate models of the requirements were used in all of the MCDM formulations. This ensured that the example problem would be executed similarly to a large-scale design problem (normally plagued with long execution times). These surrogate models form the objectives for the compromise programming algorithm, with the target values set at the positive ideal solution (always at the extreme of the equation). Each of the response surfaces are normalized by their mean and variance, so they can never result in a value higher than 1.0 or lower than -1.0. They are also normalized such that all are "larger is better," hence the deflection and weight equations are switched in sign. This is simply a matter of convention. The CP algorithm utilized the objective function from equation (8) and minimized with a built-in Matlab function utilizing Sequential Quadratic Programming (SQP) algorithm. SQP is a powerful gradient-based optimizer and is discussed at length in numerical optimization literature [Vanderplaats, 1999].
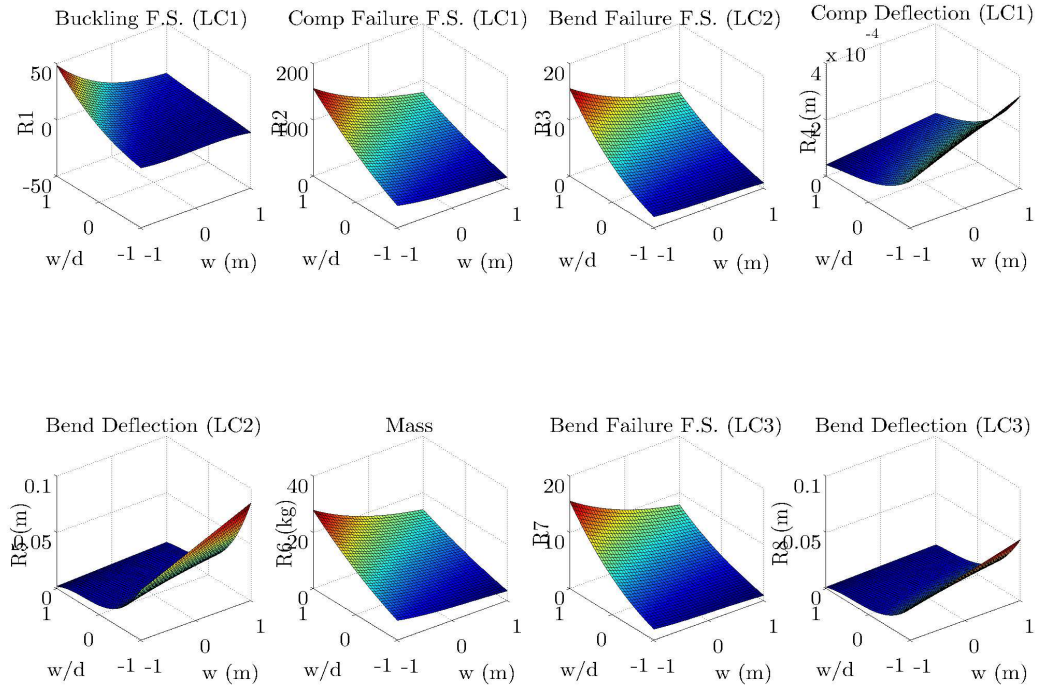
Figure 18: Surrogate Models of Beam Requirements
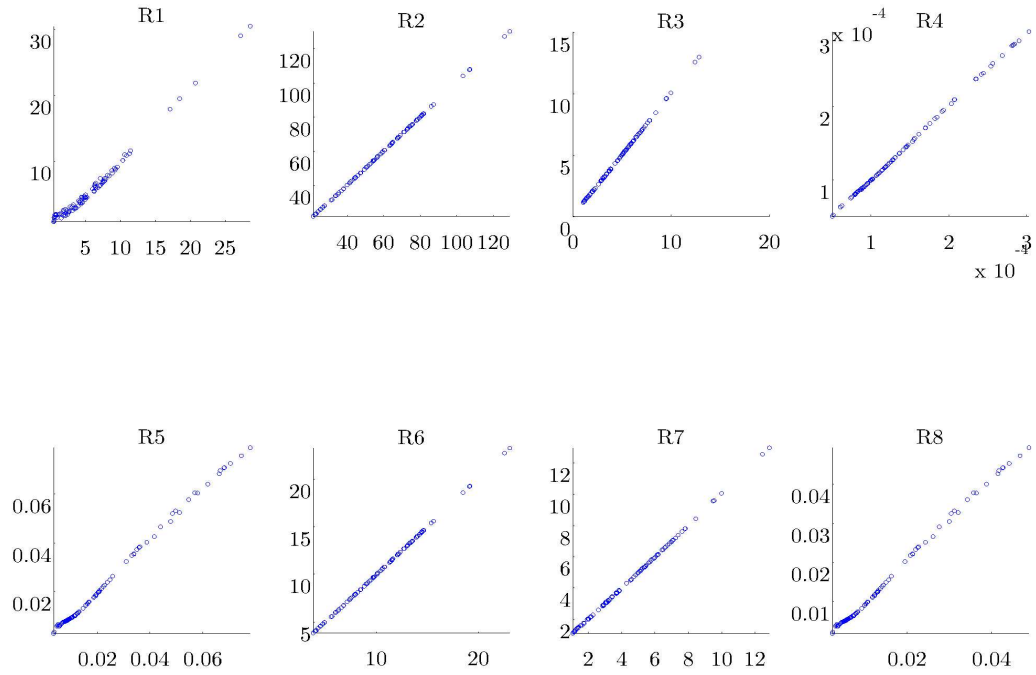
Figure 19: Actual versus Predicted Results for Surrogate Models

4.2.2 Initial Results

The initial results for the beam design problem involved using a compromise programming algorithm with evenly weighted objectives, such that the relative weights added to one. The CP algorithm was first tested with all eight requirements, then again only with requirements one through six (representing only loading conditions one and two). These results were compared to a "control" experiment of typical constrained single-objective optimization. In this case the single objective was weight and the constraints were set at 1.5 for all of the factors of safety and and 0.1 meters for all of the deflection requirements. A comparison of these three methods is presented in Table 3.

Table 3: Single and Multi-Objective Results for Beam Design Problem

Value	Single Objective Constrained: 6 Requirements	Single Objective Constrained: 8 Requirements	Compromise Programming: 6 Requirements	Compromise Programming: 8 Requirements	Units
AR	1.865	1.865	1.000	1.119	-
w	0.0558	0.0558	0.0920	0.0990	m
R_1	1.50	1.50	36.00	35.86	-
R_2	26.45	26.45	132.27	137.54	-
R_3	1.50	1.50	12.19	13.62	-
R_4	0.000249	0.000249	5.101E-05	4.670E-05	m
R_5	0.05603	0.05603	0.00583	0.00309	m
R_6	4.72	4.72	23.61	24.55	kg
R_7	N/A	1.500	N/A	13.617	-
R_8	N/A	0.03502	N/A	0.00193	m

Several interesting observations result from this table. The multi-objective solutions are far larger, heavier, and overdesigned than the single-objective constrained solution. This is largely because all of the objectives are opposed in direction of importance in some way to the beam weight objective. Since all of these objectives are evenly weighted (even relative importance), the assumption of monotonically increasing utility now makes minimizing beam weight much less important than before. Beyond this, the two compromise programming solutions are different for the six- and eight-requirement cases. This contrasts engineering intuition, which would normally have designed both beams the exact same and indicates a potential issue in the objective functions of the CP formulation. The objective functions for the six- and eight-requirement CP cases is shown in Figures 20 and 21, respectively.

The difference in these objective functions is a result of the "double-weighting" of the bending requirements. Though the intuitive solution for both is the same, the compromise programming solutions are different. The current formulation does not support the choice of "requirements groups" or other "characteristic directions" amongst dependent requirements and is thus prone to such errors. The creation of such characteristic requirements could follow from decomposition of the original requirements. These characteristic requirements would be a subset of the originals and would enable tradeoffs amongst key metrics.

Both objective functions also show a highly undesirable region near the minimum weight solution and a preference for higher-weight solutions. Certainly, the minimum-weight solution may not be the best compromise alternative, but the assumption of monotonically
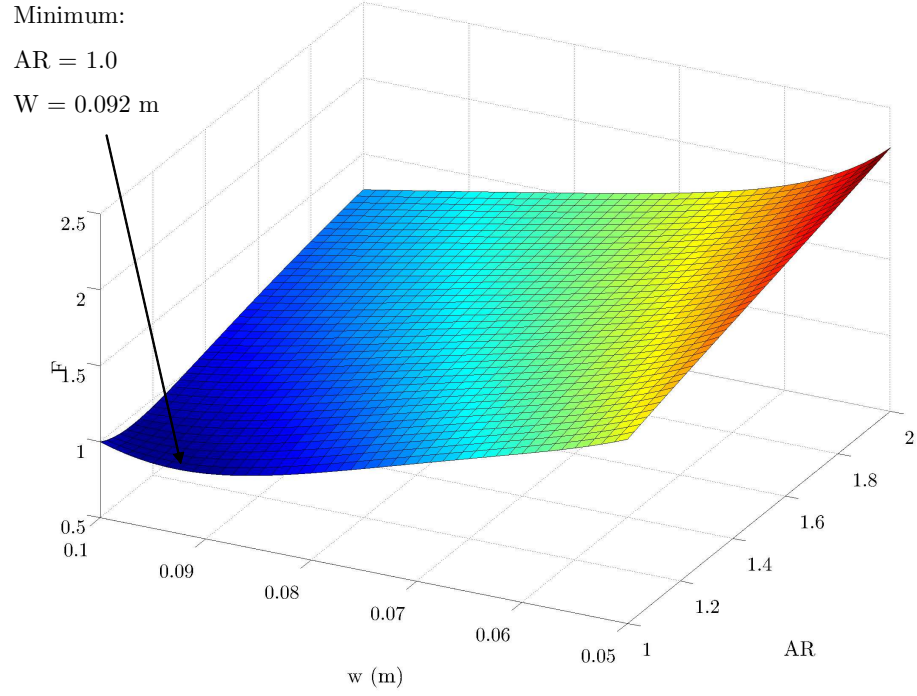
Figure 20: Objective Function of Compromise Programming Solution to Six-Requirement Beam Design Problem
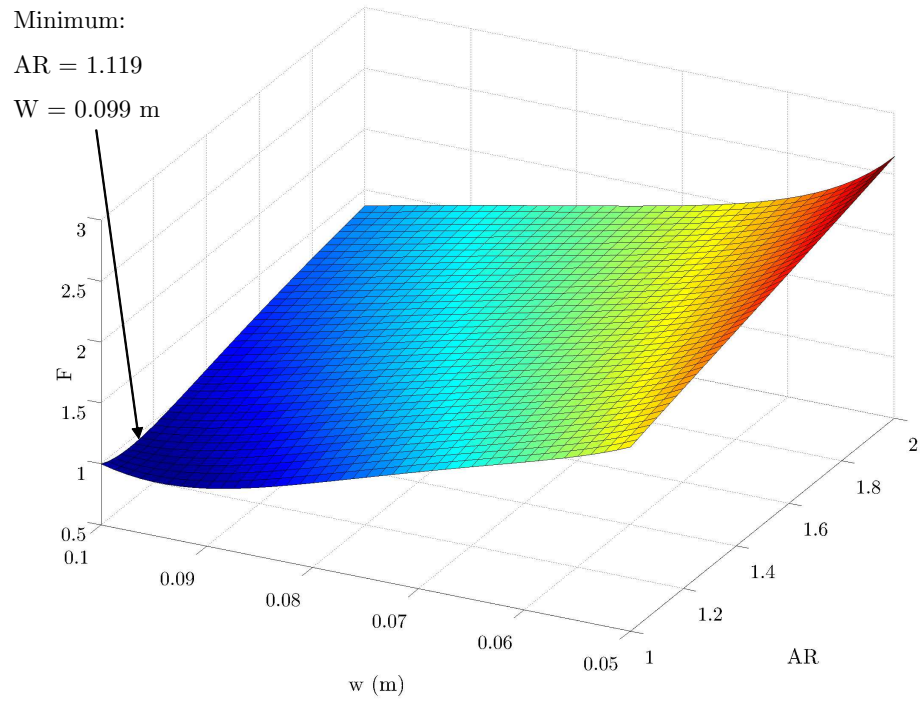
Figure 21: Objective Function of Compromise Programming Solution to Eight-Requirement Beam Design Problem

increasing utility may not hold beyond a "threshold" value of the other requirements. The current formulation does not handle threshold or constraint values in the weighting scheme, instead having a static value for relative importance.

Two pertinent hypotheses can be formulated to better reflect the decision-makers preference with respect to relative importance, threshold and constraint values, and the key tradeoffs in the decision process.

(H5) Dynamic relative importance weights better reflect the decision maker's preference for constraints and threshold values.

(H6) Key tradeoffs can be identified and ranked through decomposition of the original requirements. The best tradeoff is the solution closest to a theoretical "ideal tradeoff" in this subspace.

4.3 Weighting Schemes

The choice of the relative importance of each of the decision metrics is of paramount importance to virtually every MCDM technique. As mentioned above, the relative importance of each metric should be dynamic to mimic the decision maker's preference for constraints and threshold values. Therefore, a two-part model for relative importance weights is suggested. This two-part model in its basic form is

$$w_j = w_j^s + w_j^d(\vec{x}) \tag{21}$$

where w_j is the relative importance ("weight") of the jth metric, w_j^s is the static importance of the metric, and $w_j^d(\vec{x})$ is the dynamic importance. The static importance is a reference value given by the decision maker while the dynamic importance modifies the static value to reflect the distance of the metric from the constraint and threshold values. Note that the dynamic value is a function of the design variables as it requires information on the current value of the constraint. Uncertainty in the requirements can be captured with probability distributions about the static weight.

Some requirements may be closely related to others. If requirements are "grouped" in any way, this would modify the overall relative importance given by equation (21). Requirements grouping will be covered in a later section.

4.3.1   Entropy-Based Weights

The choice of static weights for individual metrics can follow from basic systems engineering concepts. These values are usually chosen by a qualified individual or panel of experts that debate the importance of one particular metric to another. However, it becomes increasingly difficult to assign importance to individual requirements as their number and diversity increases. Therefore, a decision maker needs some sort of aid when choosing static weights.

One promising concept is the idea of entropy-based weights. This formulation measures a quantity analogous to the "entropy" of the alternatives with respect to the decision metrics. If all of the alternatives have the same value in one of the decision metrics, its "entropy" is maximized (much like the thermodynamic concept of entropy, where equilibrium is the state of maximum entropy). If a metric is at maximum "entropy" it is of no importance to the decision maker since each of the alternatives are indifferent. Alternatively, a metric well below its maximum entropy implies that the alternatives are highly differentiated and

therefore have a higher relative importance weight. It follows that the relative importance weight of a given metric is inversely proportional to its entropy.

Zeleny [Zeleny, 1982] gives a formulation for entropy-based weighting for a discrete pool of alternatives. The procedure is as follows, using the notation introduced in Chapter III. Recall that a vector $\overline{y}_j = (\overline{y}_{1j}, \overline{y}_{2j}, \ldots, \overline{y}_{ij})$ characterizes the set $\overline{Y}$ in terms of the jth normalized attribute for i alternatives. Then define

$$\overline{Y}_j = \sum_{i=1}^{m} \overline{y}_{ij} \qquad j = 1, 2, \ldots, n \tag{22}$$

recalling that there are m total alternatives and n total metrics. The *entropy* (or its inverse, *contrast intensity*) of the jth metric is measured as

$$e(\overline{y}_j) = -K \sum_{i=1}^{m} \frac{\overline{y}_{ij}}{\overline{Y}_j} \ln \frac{\overline{y}_{ij}}{\overline{Y}_j} \tag{23}$$

where $K > 0$, ln denotes the natural logarithm, and $0 \leq \overline{y}_{ij} \leq 1$ and $e(\overline{y}_j) \geq 0$. If all $\overline{y}_{ij}$ become identical for a given j, then $\frac{\overline{y}_{ij}}{\overline{Y}_j} = \frac{1}{m}$, and $e(\overline{y}_j)$ assumes its maximum value, $e_{max} = \ln m$. Thus, setting $K = \frac{1}{e_{max}}$ will ensure that all of the $e(\overline{y}_j)$ values will fall between zero and one. This normalization helps for comparative purposes.

The *total entropy* of $\overline{Y}_j$ is defined as

$$E = \sum_{j=1}^{n} e(\overline{y}_j) \tag{24}$$

The relative importance weight of a metric is inversely proportional to its entropy. The entropy-based weights are then defined as

$$\lambda_j = \frac{1}{n - E}[1 - e(\overline{y}_j)] \tag{25}$$

where λ_j is the entropy-based weight of the jth metric. Equation (25) is normalized such that each individual entropy-based weight is between zero and one and the sum of all of the weights equals one.

If necessary, the decision maker may still modify the entropy-based weight with a preference. This makes the final static weight value defined as

$$w_j^s = \frac{w_j^{pref} \lambda_j}{\sum_{j=1}^{n} w_j^{pref} \lambda_j} \tag{26}$$

where w_j^{pref} is the value of the relative importance weight chosen by the decision maker. Note that in the absence of other strong preference information, this value is usually one.

The beam design problem introduced in this chapter has a continuous decision space that is not initially well-suited to the development of entropy-based weights. However, a sufficiently large random population may be able to produce entropy-based weights for each of the metrics. Table 4 lists the weights generated for the six- and eight-requirement beam design problem. These numbers vary slightly for each trial as they rely on an initial random population. In this case, the population contained 2,000 uniformly random alternatives.

Table 4: Entropy-Based Weights Generated for Beam Design Problem

Value	6 Requirements	8 Requirements
w_1	0.3403	0.2538
w_2	0.2334	0.1741
w_3	0.3079	0.2297
w_4	0.0418	0.0311
w_5	0.0328	0.0245
w_6	0.0438	0.0327
w_7	N/A	0.2297
w_8	N/A	0.0245

The results here are curious. The most important requirements, in descending order, appear to be R_1, R_3, R_7, and R_2 for the eight-requirement case, with R_3 and R_7 each with identical values. This is because these are the completely redundant bending stress requirements for loading conditions two and three. Note that R_6, the requirement for minimum weight, has a very low value of relative weight, an order of magnitude less than R_1, the Euler buckling requirement. Though well-intentioned, this may not reflect the decision maker's true preference.

Once again, the issue of monotonically increasing utility plays somewhat in the creation of the entropy-based weights. Functions like Euler buckling factor of safety given in equation (13) have the most curvature in the decision space, and therefore will have the lowest entropy simply due to the large differentiation. In a sense, the entropy of each attribute is related to the difference between the mean and extreme values of each function. Table 5 lists the ranges of each requirement to illustrate the origins of the entropy-based weights.

Table 5: Ranges of Requirements for Beam Design Problem

Value	Best	Worst	Units
R_1	50.102	0.391	-
R_2	156.330	19.542	-
R_3	15.633	0.977	-
R_4	4.10E-05	0.00032832	m
R_5	0.00274	0.08755	m
R_6	3.4875	27.9	kg
R_7	15.633	0.977	-
R_8	0.00171	0.05472	m

This problem can be partially alleviated again with the introduction of threshold values. If the decision maker has already accepted the concept of a threshold value (discussed in greater detail in the next section), it too would have an effect on the entropy-based weight. In effect, the threshold value indicates indifference amongst concepts beyond the threshold. All values beyond the threshold should be treated as values *in equilibrium* as far as the decision maker is concerned. Hence, all alternatives above the threshold can be replaced

in the entropy calculations as having the same value, hence *raising* the entropy of that particular metric. This would serve to reduce the contrast intensity and therefore relative importance of the affected metric.

Consider again the beam problem with the addition of threshold values. Here, the decision maker may decide that factors of safety beyond 3.0 and deflections smaller than 0.05 meters are thresholds for this particular problem, with no threshold on weight. In essence, this states that if all the threshold values for an alternative are exceeded this becomes a single-objective optimization problem for weight only. Table 6 lists the entropy-based weights that account for threshold values. Note that these are still the *static weights*, and that thresholds also play an important part in the *dynamic weights* of the environment.

Table 6: Modified Entropy-Based Weights for Beam Design Problem Considering Threshold Values

Value	6 Requirements	8 Requirements
w_1	0.1020	0.0699
w_2	3.12E-12	2.14E-12
w_3	0.4507	0.3088
w_4	3.12E-12	2.14E-12
w_5	0.0839	0.0575
w_6	0.3634	0.2490
w_7	N/A	0.3088
w_8	N/A	0.0060

The modified entropy-based weights seem to make much more sense to the decision maker. The importance weights for R_2 and R_4 are essentially zero, reflecting the fact that both of these metrics are *always* greater than their threshold values. This resonates with the original engineering intuition that the compressive failure and deflection cases will not be design drivers, and indeed in this design space are inconsequential. Further, the importance of R_1 is greatly reduced because much of this space is beyond the threshold value for buckling factor of safety. The two most important requirements become bending failure (R_3 and R_7) and beam weight (R_6). However, double-weighting is still quite possible for bending in the eight-requirement case. Further, this does not handle the portions of the space that are actually below the constraints. This could potentially result in a solution that does not meet constraints, or is capable of "over-optimizing" beyond threshold values, though the latter case can happen only if the threshold is within the decision space. These effects need to be captured with the dynamic weights.

4.3.2   Constraint and Threshold Modeling

The lack of internal constraints and threshold values in the generalized MCDM formulation allows the engineer direct control over these values. Furthermore, it embodies the basic approach in probabilistic MCDM: elimination of as many constraints as possible to enable tradeoffs. Many requirements that are considered "constraints" are actually arbitrary performance levels set forth by a committee well before system definition and synthesis are performed. These performance levels may be set to dominate performance of an existing concept, as a minimum goal for performance, or sometimes simply to help define the

problem better for typical single-objective optimization or fuel-balance sizing. Elimination of these constraints is crucial to MCDM as it allows for implicit tradeoffs to be made on high-payoff objectives at the expense of less important objectives. However, some actual constraints do exist, usually for safety, certification, or resource reasons. Several examples of true constraints can be seen in meeting the Code of Federal Regulations for civil transport design [Federal Aviation Administration, 2005]. As such, the decision maker does need to be able to input constraints into the MCDM environment.

The concept of threshold values was first introduced in the static weighting scheme. These are even more important for the dynamic portion of the weights if the threshold lies somewhere within the decision space. Beyond the threshold, improvement in the metric becomes inconsequential. Modeling this behavior is crucial if the decision maker has a threshold in mind.

In a basic sense, the ideal dynamic weight will rise sharply if its associated constraint is violated. This makes it the most important objective in the space until the constraint is satisfied. At this point, the dynamic weight quickly drops to zero and the MCDM algorithm proceeds using only the static weight. If the metric improves much beyond its constraint and has a threshold value the dynamic weight will become slightly negative, becoming more negative until it reaches the threshold value, upon which it is equal to exactly the opposite of the static weight value. Therefore, the sum of the static and dynamic weights equals exactly zero at or beyond the threshold, effectively negating any influence of that metric on the MCDM algorithm.

Specifically, this dynamic weight cannot have too sharp of a change in value. If it were too sharp it could increase the difficulty associated with gradient-based optimizers. Therefore, continuous, smooth functions should be used whenever possible. Further, these dynamically modified weights should *not* be re-normalized to sum to one, else any change in one weight would change all others. This could also have an adverse effect on gradient-based optimizers.

The constraint portion of the dynamic weight model should approximate a step function. One attractive option is to find a smooth approximation of the Heaviside Step Function, also known as the Unit Step Function [Weisstein, 2005]. An approximation to this is defined as

$$w_j^{dc}(\vec{x}) = -1\left[-\frac{1}{2} + \frac{1}{\pi}\tan^{-1}\left[s(\overline{y}_j(\vec{x}) - \overline{c}) \right] \right] \tag{27}$$

where w_j^{dc} refers to the constraint contribution to the jth dynamic weight, $\overline{y}_j(\vec{x})$ refers to the current value of the jth metric, $\overline{c}$ refers to the normalized constraint value, and s is a scale factor for controlling smoothness of the step function. Larger values of s will make the step function more abrupt. If s is too abrupt there may be problems with gradient-based optimizers, whereas if it is too small it will not jump fast enough near the constraint. A good value of s is approximately 200, though this is best tailored to the individual problems and optimization algorithms used.

The threshold value needs to have a more gradual decay, such as that seen in an inverse power relationship. It should be zero at the constraint value (or the edge of the design space, if no constraint is present) and should be opposite of the static weighting value at the threshold value. A good model for this is

$$w_j^{dt}(\vec{x}) = -w_j^s\left(1 - \frac{\overline{t} - \overline{y}_j(\vec{x})}{\overline{t} - \overline{c}} \right)^{\frac{1}{4}} \tag{28}$$

41

where w_j^{dt} is the threshold contribution to the jth dynamic weight and $\bar{t}$ is the normalized threshold value. Note that both $\bar{c}$ and $\bar{t}$ should be normalized by the same procedure as response surface equations; that is, by the mean and variance of their respective responses. If $\bar{c}$ is less than -1.0 then the entire decision space is beyond the constraint; likewise for $\bar{t}$. If either value is greater than 1.0 it implies that the entire space is smaller than the constraint or threshold value.

Equations (27) and (28) can be combined into a single dynamic weight that is valid across the entire space no matter what its nature. This is represented by

$$
w_j^d(\vec{x}) = \begin{cases} \left[\frac{1}{2} - \frac{1}{\pi}\tan^{-1}\left[s(\bar{y}_j(\vec{x}) - \bar{c})\right]\right] - w_j^s\left(1 - \frac{\bar{t}-\bar{y}_j(\vec{x})}{\bar{t}-\bar{c}}\right)^{\frac{1}{4}} & \bar{y}_j(\vec{x}) \leq \bar{t} \\ -w_j^s & \bar{y}_j(\vec{x}) > \bar{t} \end{cases} \tag{29}
$$

with the notation as noted before. A plot of an example dynamic weight function for all $y_j(\vec{x})$ is given in Figure 22.
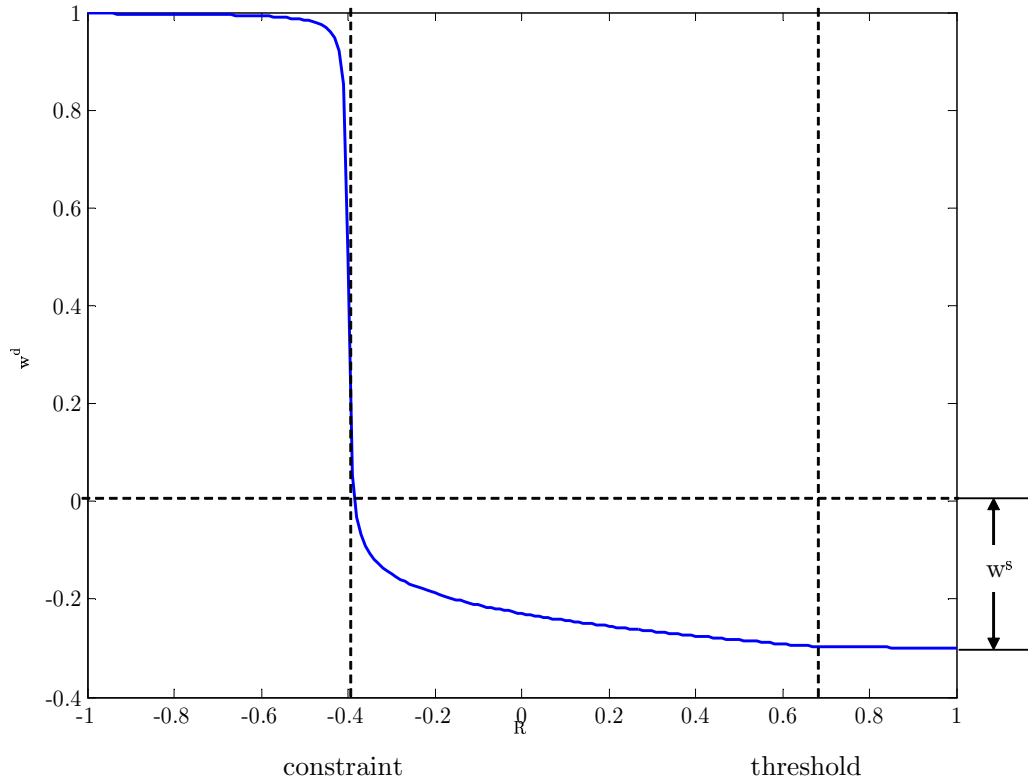
Figure 22: Example Dynamic Weight Variation with $y_j(\vec{x})$

4.3.3   Beam Design Problem with Dynamic Importance Weighting

The example beam design problem was reformulated with all of the concepts discussed in this section: entropy-based static weights with thresholds, augmented with dynamic weights with appropriate constraints and thresholds. The threshold values were as discussed earlier; all factors of safety were considered indifferent above 3.0 and all deflections below

0.05 meters. The constraints chosen reflected those common in aerospace design, with constraints for factors of safety set at 1.5 and constraints for deflections set to greater than 0.1 meters. The response for beam mass was not constrained, nor did it contain a threshold value. The overall weights for all of the requirements were found from the sums of the entropy-based weights given in Table 6 and the appropriate dynamic weights. These are shown as a function of $y_j(\vec{x})$ in Figure 23. The weights are shown on a logarithmic scale for greater contrast.
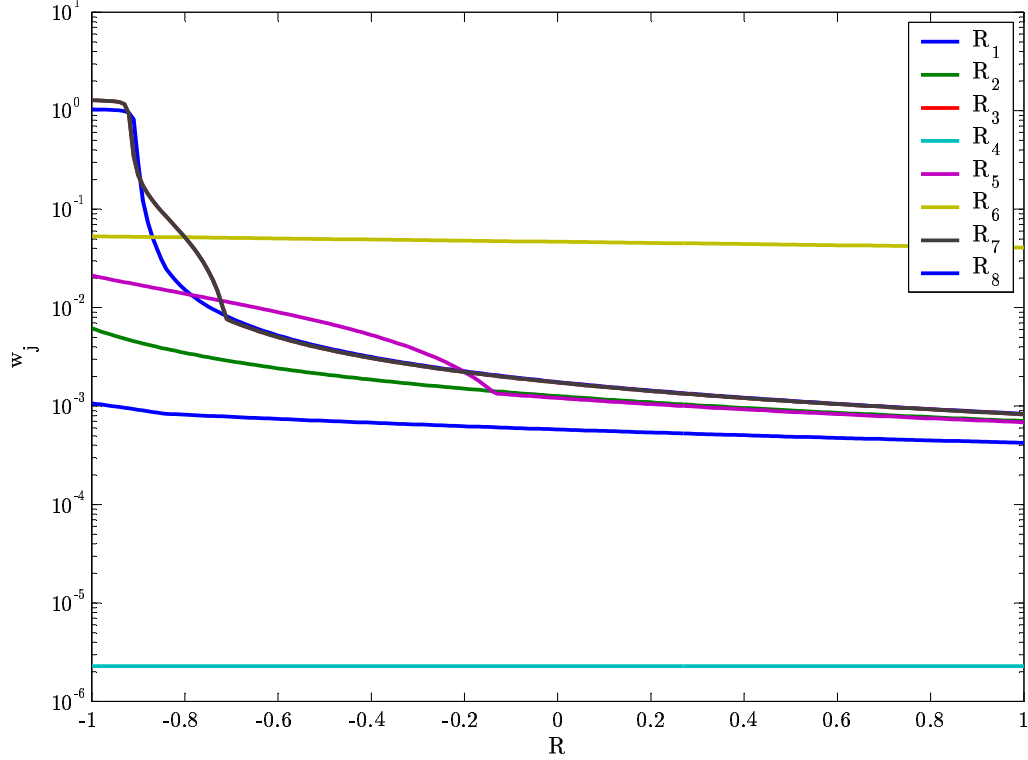
Figure 23: Variation of Weights for Beam Design Problem with $y_j(\vec{x})$

Note that some of the responses are well beyond their threshold values (as before) so they exhibit no dynamic behavior and are stuck around zero. Other values, such as beam mass, have no dynamic behavior because of lack of thresholds. Still others have regions of influence and show spikes near their respective constraints and decay towards their threshold values. Also interesting is the plots of the modified CP objective functions of the six- and eight-requirement problem with these dynamic weights. Both of the objective functions are still physically different but now give virtually the same answer due to the threshold dropoffs. These objective functions are illustrated in Figures 24 and 25 and the CP answers for both cases are presented in Table 7. These results are promising for large-scale MCDM, but still do not fully address the issues associated with double-weighting. The idea of requirements decomposition addresses this next.
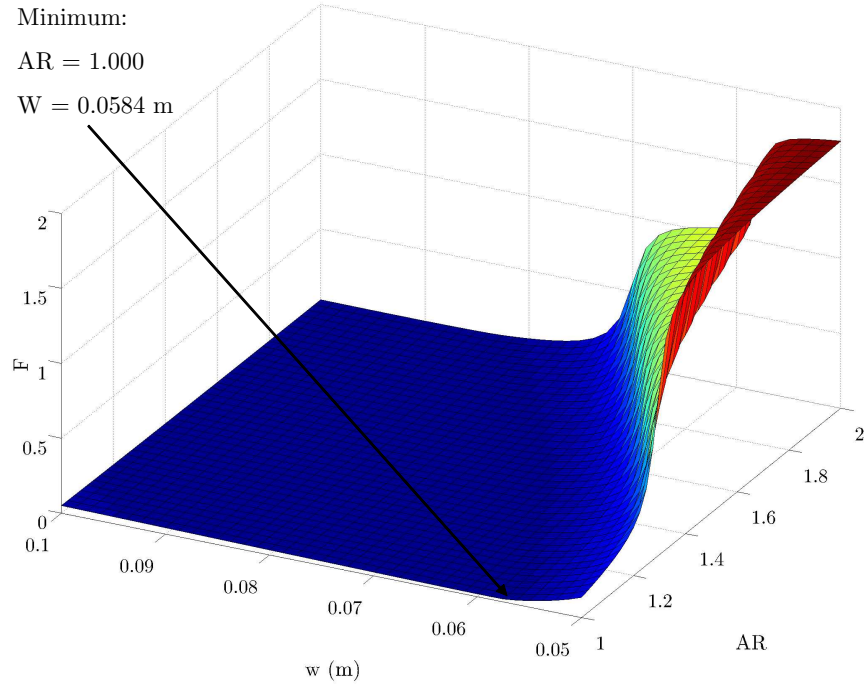
Figure 24: Objective Function for Six-Requirement Beam Design Problem with Dynamic Weights
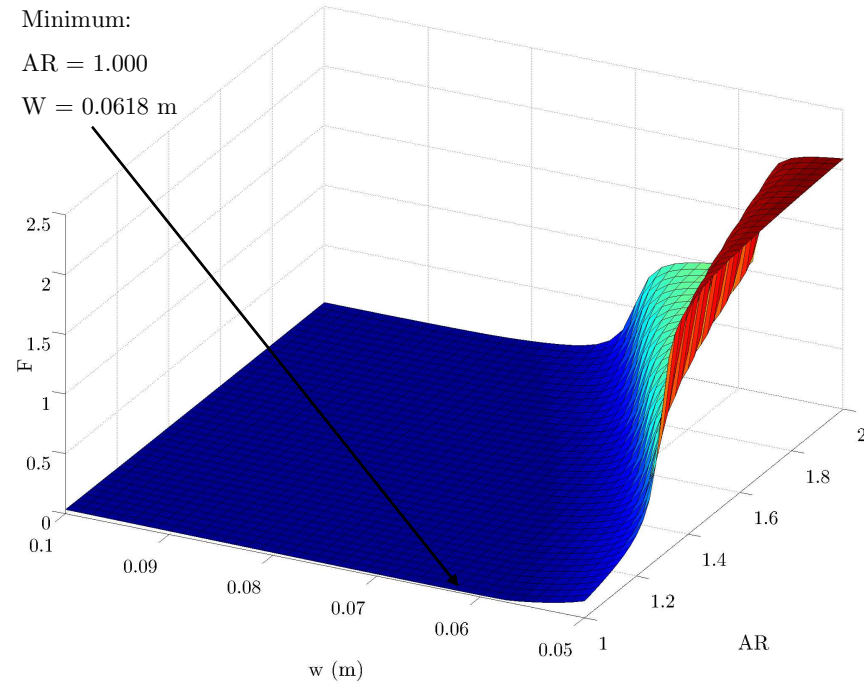
Figure 25: Objective Function for Eight-Requirement Beam Design Problem with Dynamic Weights

Table 7: Compromise Programming Results for Beam Design Problem with Dynamic Weights

Value	6 Requirements	8 Requirements	Units
AR	1.000	1.000	-
w	0.0584	0.0618	m
F*	0.0169	0.0139	-
R_1	5.945	7.557	-
R_2	53.56	59.90	-
R_3	3.125	3.703	-
R_4	0.000119	0.000106	m
R_5	0.02331	0.01790	m
R_6	9.56	10.69	kg
R_7	N/A	3.703	-
R_8	N/A	0.01119	m

4.4 Requirements Decomposition

The idea of requirements decomposition (or compression) is not new. The principles of Axiomatic Design [Suh, 1990], detailed in Chapter III, discuss the need for a minimal set of independent functional requirements. The relation between functional requirements and design parameters can be represented by

$$
\underbrace{\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix}}_{\vec{R}} = \underbrace{\begin{bmatrix} \frac{\partial R_1}{\partial x_1} & \frac{\partial R_1}{\partial x_2} & \cdots & \frac{\partial R_1}{\partial x_q} \\ \frac{\partial R_2}{\partial x_1} & \frac{\partial R_2}{\partial x_2} & \cdots & \frac{\partial R_2}{\partial x_q} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial R_n}{\partial x_1} & \frac{\partial R_n}{\partial x_2} & \cdots & \frac{\partial R_n}{\partial x_q} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix}}_{\vec{x}}
\tag{30}
$$

where $\vec{R}$ refers to a column vector of n requirements (metrics), $\mathbf{B}$ refers to a $q \times n$ matrix of partials, and $\vec{x}$ is a column vector of q design variables. In perfect Axiomatic Design, the $\mathbf{B}$ matrix is square and all of the off-diagonal terms are zero; that is, the functional requirements are independent and are controlled by only one design parameter.

Unfortunately, independence of the functional requirements is *very* hard to enforce and becomes all but impossible for large-scale systems. Zeleny [Zeleny, 1982] gives an example with dependent requirements and notes some of the issues associated with creation of composite attributes:

> The problem is that [as] the number of attributes increases . . . such composite attributes are often difficult to quantify and even to conceptualize.

Furthermore, the off-diagonal terms in $\mathbf{B}$ can be thought of as interactions between the requirements. These interactions, carefully presented, are often the crux of design, decision making, and compromise. All but the simplest systems will be compromised; else, the ideal solution would always be possible. However, this is not to say that a *linearly* independent set of requirements is not desirable for MCDM. If one believes the set of existing requirements

helps to "paint a picture" of the decision space, then these requirements should be able to be decomposed into their cardinal directions for a linearly independent, though interacting, set of *characteristic requirements.*

One unrealized advantage for the use of polynomial surrogate models is that the approximate functional form of the requirements to the design parameters is known. That is, a polynomial response surface equation is an approximation for the terms in $\mathbf{B}$. Therefore, if the partials in $\mathbf{B}$ are linearly dependent row-wise, there exists a linearly independent subset of partials that can be combined to form $\vec{R}$.

4.4.1 Decomposition Techniques for Linear Systems

One of the most fundamental decomposition techniques for matrices is diagonalization utilizing eigenvalues and eigenvectors [Strang, 1988; Nakos and Joyner, 1998]. A square matrix can be represented as the product of three orthogonal matrices via

$$\mathbf{A} = \mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1} \tag{31}$$

where $\mathbf{A}$ is a diagonalizable square matrix, $\mathbf{S}$ is a matrix of eigenvectors of $\mathbf{A}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix of the associated eigenvalues. The rows of $\mathbf{S}^{-1}$ form an orthogonal basis for the row space of $\mathbf{A}$; that is, these rows represent a linearly independent set of vectors. If $\mathbf{A}$ is singular, some of the eigenvalues in $\boldsymbol{\Lambda}$ will be zero and therefore the associated eigenvectors will be unimportant to the recomposition of $\mathbf{A}$. Similarly, if $\mathbf{A}$ is nearly singular, there will be some eigenvalues that will be nearly zero. The associated eigenvectors will be of little importance to the recomposition and may be neglected with little error. This procedure is used in vibrational analysis to determine the important natural modes of a structure [James et al., 1994].

Unfortunately, it is very rare that the number of requirements equals the number of design variables, so a square matrix is quite rare. One technique that is comparable to eigenanalysis is Singular Value Decomposition (SVD). This is a factorization method that retains some of the qualities of eigenanalysis except is applicable to non-square matrices. In its basic form, SVD is written as

$$\underbrace{\mathbf{A}}_{a \times b} = \underbrace{\mathbf{U}}_{a \times a} \underbrace{\boldsymbol{\Sigma}}_{a \times b} \underbrace{\mathbf{V}^T}_{b \times b} \tag{32}$$

where $\mathbf{A}$ is now a rectangular $a \times b$ matrix of rank c, $\mathbf{U}$ and $\mathbf{V}^T$ are orthonormal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix of *singular values.* SVD orders the columns of $\mathbf{U}$ and $\mathbf{V}^T$ such that each column of the former and row of the latter correspond to the singular values on the diagonal of $\boldsymbol{\Sigma}$ in descending order. Thus, the most important columns and rows of $\mathbf{U}$ and $\mathbf{V}^T$ appear first. Further, the first c columns of $\mathbf{U}$ form an orthonormal basis for the column space of $\mathbf{A}$. Likewise, the first c rows of $\mathbf{V}^T$ for an orthonormal basis for the row space of $\mathbf{A}$. SVD is very stable and, as its name implies, works well with singular or near-singular matrices. In the case of near-singular matrices, the singular values along the diagonal of $\boldsymbol{\Sigma}$ will have some near-zero or zero values. The corresponding columns of $\mathbf{U}$ or rows of $\mathbf{V}^T$ are therefore unimportant to the recomposition of $\mathbf{A}$ and can be neglected with little loss in accuracy. For more information on SVD, the reader is referred to [Strang, 1988].

4.4.2 Singular Value Decomposition for Response Surface Equations

Response surface equations are already of great use to the systems engineer because of their speed, versatility, and high-fidelity capability. The general form of a quadratic RSE is

$$R = b_0 + \sum_{i=1}^{q} b_i x_i + \sum_{i=1}^{q} b_{ii} x_i^2 + \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} b_{ij} x_i x_j \tag{33}$$

where b_0 is an intercept, b_i are the coefficients for main (linear) effects, b_{ii} are coefficients for the quadratic effects, b_{ij} are coefficients for interaction terms, and x_i or x_j refers to one of q design variables. One can assemble all of the coefficients via

$$\vec{b} = (b_0, b_1, b_2, \ldots, b_q, b_{11}, b_{12}, \ldots, b_{1q}, b_{22}, b_{23}, \ldots, b_{qq})^T \tag{34}$$

such that all coefficients can be represented by a single vector. Likewise, $\vec{x}$, the vector of design variables, can be rearranged via

$$\vec{\tilde{x}} = (1, x_1, x_2, \ldots, x_q, x_1 x_1, x_1 x_2, \ldots, x_1 x_q, x_2 x_2, x_2 x_3, \ldots, x_q x_q)^T \tag{35}$$

such that $R = \vec{b}\vec{\tilde{x}}$. Then, for multiple response surface equations, the responses can be arranged such that

$$\vec{R} = \mathbf{B}\vec{\tilde{x}} \tag{36}$$

where $\vec{R} = (R_1, R_2, \ldots, R_n)^T$ is a vector of n responses and $\mathbf{B} = (\vec{b_1}, \vec{b_2}, \ldots, \vec{b_n})^T$ is a matrix of RSE coefficients for the n responses. Note the similarity of equation (36) to the basic design formulation given in equation (30). The chief differences are the allowance of quadratic terms in the model to capture additional nonlinear effects.

Now, each row of $\mathbf{B}$ represents the coefficients for a separate response surface equation. If $\mathbf{B}$ is singular or is close to being singular, then some of the response surface equations, and hence the responses themselves, are linearly dependent. This can be seen from singular value decomposition of the $\mathbf{B}$ matrix:

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{37}$$

In this case, $\mathbf{B}$ is an $n \times v$ rectangular matrix, where v is the length of $\vec{\tilde{x}}$ (from equation (35) and represents the number of terms in the response surface. The rank of $\mathbf{B}$ depends on its shape; usually, $n < v$ so the matrix will be at most of rank n. Note that if $n > v$ then there will *always* be linear dependence amongst the responses, and the matrix will be at most of rank v. For now, the reader can assume that the rank of the matrix will be n, in cases where it is not a simple substitution will be required in subsequent equations.

Assuming $n < v$, $\mathbf{B}$ is at most of rank n. Therefore, $\mathbf{U}$ is an $n \times n$ orthonormal matrix, $\mathbf{\Sigma}$ is an $n \times v$ diagonal matrix, and $\mathbf{V}^T$ is a $v \times v$ orthonormal matrix. Furthermore, the first n rows of $\mathbf{V}^T$, denoted $\mathbf{V}_1^T$, form an orthonormal basis for the row space of $\mathbf{B}$. Therefore, a *linearly independent* set of RSEs that can be used to describe the same decision space given by the original RSEs is found from

$$\vec{R}' = \mathbf{V}_1^T \vec{\tilde{x}} \tag{38}$$

where $\vec{R}'$ represents the equations for the *characteristic requirements* of the system. These requirements have no associated direction of improvement on their own but form a series

of independent equations used to model the decision space. The n singular values give the importance of each of the associated characteristic requirements. Thus, a near-zero singular value indicates that the particular characteristic requirement associated to that singular value does not describe much of the variation seen in the decision space (made up of all of the original requirements) and therefore can be removed from consideration without impacting the decision making process. This is key in determining both the important characteristic requirements and reducing the size of the decision space.

This procedure can be illustrated on the beam design problem. The normalized RSE coefficients for $n = 8$ requirements are given in Table 8. Note that in this case the RSE is a third-order model; therefore, there are $v = 10$ terms in each model, making $\mathbf{B}$ an 8×10 matrix.

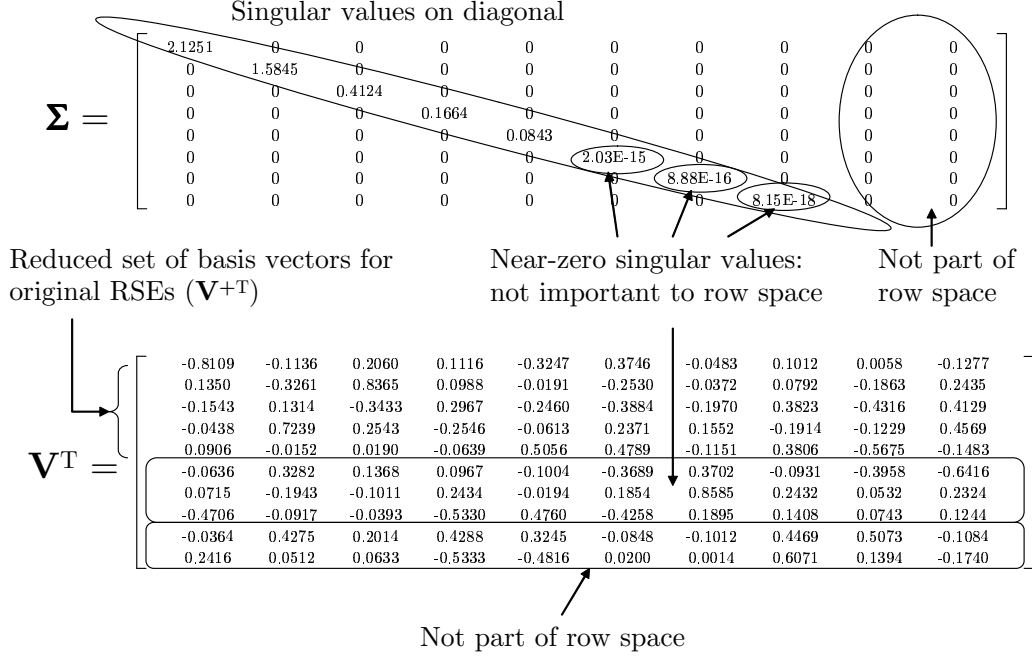**Table 8:** Third-Order Normalized Response Surface Equation Coefficients for Beam Design Problem

Response	Term									
	intercept	AR	w	AR^2	AR x w	w^2	AR^3	AR^2 x w	AR x w^2	w^3
R_1	-0.8109	-0.1572	0.2350	0.2359	-0.3896	0.1855	-0.1287	0.2571	-0.1816	0.0404
R_2	-0.4264	-0.2831	0.5707	0.1145	-0.2107	0.1011	-0.0388	0.0723	-0.0351	0.0000
R_3	-0.5295	-0.1959	0.5983	0.0887	-0.2283	0.2125	-0.0300	0.0784	-0.0738	0.0236
R_4	0.5232	-0.2516	0.5024	0.0000	0.2073	-0.3158	0.0000	0.0000	-0.1053	0.1407
R_5	0.7628	-0.0940	0.3841	0.0000	0.2251	-0.5377	0.0000	0.0000	-0.1792	0.3427
R_6	0.4264	0.2831	-0.5707	-0.1145	0.2107	-0.1011	0.0388	-0.0723	0.0351	0.0000
R_7	-0.5295	-0.1959	0.5983	0.0887	-0.2283	0.2125	-0.0300	0.0784	-0.0738	0.0236
R_8	0.7628	-0.0940	0.3841	0.0000	0.2251	-0.5377	0.0000	0.0000	-0.1792	0.3427

From here, SVD is applied to the $\mathbf{B}$ matrix. The values found from this analysis for $\boldsymbol{\Sigma}$ and $\mathbf{V}^T$ are shown in Figure 26. This figure depicts the entire $n \times v$ matrix of singular values as well as the whole $v \times v$ matrix of coefficients for the characteristic requirements.

The singular values in Figure 26 indicate that only five characteristic requirements are of any importance to the decision space, whereas the three others have singular values of effectively zero. This perfectly coincides with the reality of the functional form of the problem: of the eight original requirements, only five are linearly independent. It further shows that only three of the characteristic requirements are very important; the other two are of marginal importance and could probably be neglected with little impact on the decision space.

Singular value decomposition of the original requirements provides exceptionally powerful results for reducing the number of descriptors of the decision space. However, some caveats are important to understand before moving on. First, normalization of the responses when building the RSEs is *extremely* important. Otherwise, responses in the natural form may be of vastly different magnitudes that will directly effect the magnitude of the singular values. Normalization by first subtracting the mean of the response followed by dividing by its variance provides a normalization between -1.0 and 1.0 which appears to be sufficient. Similarly, the RSEs must be built with normalized design variables, or the individual terms within the model may again have magnitude errors. It is also helpful to normalize to a single direction of improvement. The numbers presented in Table 8 were normalized in this fashion for a "larger is better" case. Thus, the "smaller is better" terms need simply to be multiplied by -1.0.

Singular values on diagonal

$$\boldsymbol{\Sigma} = \begin{bmatrix} 2.1251 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5845 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4124 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1664 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0843 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.03\text{E-}15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8.88\text{E-}16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8.15\text{E-}18 & 0 & 0 \end{bmatrix}$$

Reduced set of basis vectors for original RSEs ($\mathbf{V}^{+T}$)

Near-zero singular values: not important to row space

Not part of row space

$$\mathbf{V}^T = \begin{bmatrix} -0.8109 & -0.1136 & 0.2060 & 0.1116 & -0.3247 & 0.3746 & -0.0483 & 0.1012 & 0.0058 & -0.1277 \\ 0.1350 & -0.3261 & 0.8365 & 0.0988 & -0.0191 & -0.2530 & -0.0372 & 0.0792 & -0.1863 & 0.2435 \\ -0.1543 & 0.1314 & -0.3433 & 0.2967 & -0.2460 & -0.3884 & -0.1970 & 0.3823 & -0.4316 & 0.4129 \\ -0.0438 & 0.7239 & 0.2543 & -0.2546 & -0.0613 & 0.2371 & 0.1552 & -0.1914 & -0.1229 & 0.4569 \\ 0.0906 & -0.0152 & 0.0190 & -0.0639 & 0.5056 & 0.4789 & -0.1151 & 0.3806 & -0.5675 & -0.1483 \\ -0.0636 & 0.3282 & 0.1368 & 0.0967 & -0.1004 & -0.3689 & 0.3702 & -0.0931 & -0.3958 & -0.6416 \\ 0.0715 & -0.1943 & -0.1011 & 0.2434 & -0.0194 & 0.1854 & 0.8585 & 0.2432 & 0.0532 & 0.2324 \\ -0.4706 & -0.0917 & -0.0393 & -0.5330 & 0.4760 & -0.4258 & 0.1895 & 0.1408 & 0.0743 & 0.1244 \\ -0.0364 & 0.4275 & 0.2014 & 0.4288 & 0.3245 & -0.0848 & -0.1012 & 0.4469 & 0.5073 & -0.1084 \\ 0.2416 & 0.0512 & 0.0633 & -0.5333 & -0.4816 & 0.0200 & 0.0014 & 0.6071 & 0.1394 & -0.1740 \end{bmatrix}$$

Not part of row space

Figure 26: Singular Values and Coefficients of Characteristic Requirements for Beam Design Problem

Recomposition of the original requirements follows from the basics of SVD outlined in equation (37). A matrix of constants can be defined such that the characteristic requirements can be quickly converted to the originals. First, one must define the cutoff point of "important" requirements via the singular values. This cutoff point is denoted with r, such that $r \le n$. This carries over to the coefficient matrix for the characteristic requirements, with the "important" characteristics denoted by $\mathbf{V}_1^{+T}$, or simply $\mathbf{V}^{+T}$ for brevity. It follows that

$$\underbrace{\vec{R}'^{+}}_{r \times 1} = \underbrace{\mathbf{V}^{+T}}_{r \times v} \underbrace{\vec{\tilde{x}}}_{v \times 1} \tag{39}$$

where the '+' superscript denotes the first r values. The matrix of constants is defined as

$$\underbrace{\mathbf{C}^{+}}_{n \times r} = \underbrace{\mathbf{U}}_{n \times n} \underbrace{\boldsymbol{\Sigma}^{+}}_{n \times r} \tag{40}$$

with $\mathbf{C}^{+}$ as the matrix of constants to recompose the requirements using

$$\vec{\tilde{R}} = \mathbf{C}^{+} \vec{R}'^{+} \tag{41}$$

where $\vec{\tilde{R}}$ are the recomposed n original requirements. The accuracy of these values will depend on how many marginally important characteristic requirements are neglected in $\vec{R}'^{+}$, but the differences between $\vec{\tilde{R}}$ and $\vec{R}$ are usually quite small.

As seen from Figure 26, only three singular values are relatively large for the beam design problem, indicating that only three characteristic requirements are necessary to describe

the decision space. The $\mathbf{C}^+$ matrix can thus be built for this problem from the first three singular values of the $\mathbf{\Sigma}$ matrix. The $\mathbf{C}^+$ matrix for this problem is given in Table 9. Here, the columns represent the relative contributions of a particular characteristic requirement to the original requirements.

Table 9: Recomposition Coefficient Matrix for Beam Design Problem

$\mathbf{C}^+$	R'_1	R'_2	R'_3
R_1	0.9722	0.1910	0.3363
R_2	0.6235	0.5156	-0.0703
R_3	0.7445	0.4791	-0.0720
R_4	-0.4964	0.7027	-0.1111
R_5	-0.8481	0.7034	0.1104
R_6	-0.6235	-0.5156	0.0703
R_7	0.7445	0.4791	-0.0720
R_8	-0.8481	0.7034	0.1104

The $\mathbf{C}^+$ matrix can help the user determine what the characteristic requirements represent. Obviously, if a characteristic is important in the recomposition of the original requirements, the corresponding value in the $\mathbf{C}^+$ matrix will have a large magnitude. If it is not important, the magnitude will be low. Furthermore, the sign of the components of the $\mathbf{C}^+$ matrix tells a decision maker if one of the original requirements increases or decreases along a particular characteristic. If one column of $\mathbf{C}^+$ is all one sign, this indicates that all the original requirements increase or decrease with a particular characteristic, so this characteristic can be maximized on minimized appropriately. However, this will likely not be the case. More often, the signs differ significantly, requiring more advanced decision making techniques.

4.4.3 Decision Making with Characteristic Requirements

The decomposition of requirements into a subset of linearly independent characteristic requirements represents a radical departure from traditional design and decision making techniques. It is possible to combine these characteristic requirements into the originals with little to no loss in fidelity. However, decision making with the characteristic requirements is no longer a simple compromise programming maximization problem (as it is with the normalized original requirements). The characteristic requirements have no explicit direction of improvement. As an analogy, image two perfectly correlated, yet opposite, requirements. Decomposition of these two requirements will yield one characteristic requirement. One of the original requirements will become better as the value of the characteristic increases, whereas the other will become worse. What is the best compromise? The answer is a "nominal the best" optimization problem, with some sort of target value. To continue this analogy, if both requirements were equally important, and the characteristic requirement made equal contributions to both original requirements, then the target value for the "ideal tradeoff" will be a value located at exactly the middle of the characteristic.

If all of the requirements were equally important, the ideal tradeoff could be characterized from the $\mathbf{C}^+$ matrix. As mentioned before, each column vector of this matrix gives

the relative contribution of the corresponding characteristic requirement to the original requirements. If the original requirements are normalized from -1.0 to 1.0 and such that all are "larger is better," this directional information can be used to find a target value such that

$$t = \frac{(c_{1k} + c_{2k} + \cdots + c_{nk})}{|(c_{1k}| + |c_{2k}| + \cdots + |c_{nk}|)} \tag{42}$$

where c_{jk} refers to the constant in the $\mathbf{C}^+$ matrix that relates the contribution of the kth characteristic requirement to the jth original requirement. Unfortunately, it is not very often that the requirements are evenly weighted. Equation (42) can be modified for arbitrarily weighted requirements. For brevity, this can be represented in vector form as

$$t = \frac{\vec{c}_k^T \vec{w}}{|\vec{c}_k^T| \vec{w}} \tag{43}$$

where $\vec{c}_k$ is a column vector of constants relating the kth characteristic requirement to all of the original requirements and $\vec{w}$ is a column vector of arbitrary weights $(w_1, w_2, \ldots, w_n)^T$. These weights can be static or a combination of static and dynamic weights as in equation (21). The target values for all of the characteristic requirements in matrix form is then simply

$$\vec{t} = \frac{\mathbf{C}^{+T} \vec{w}}{|\mathbf{C}^{+T}| \vec{w}} \tag{44}$$

where $\vec{t}$ is a column vector of target values for the r characteristic requirements.

This target value formulation brings about a new meaning to the characteristic requirements. These "characteristics" represent the critical tradeoffs amongst the original requirements, with the singular values used to rank the importance of each of these tradeoffs. In all but cases of linear, non-interacting response surfaces and other trivial formulations, it will be impossible to meet all of the target values of the characteristics. Therefore, these target values help create a well-defined compromise programming problem. This is given as

$$\min \quad F = \left[\sum_{k=1}^{r} \left(\frac{\sigma_k (R_k' - t_k)}{-2} \right)^p \right]^{\frac{1}{p}} \tag{45}$$

where σ_k is an arbitrary scaling factor for the kth characteristic requirement, R_k' and t_k are the kth characteristic and target value, respectively, and all other notation is as in equation (8). Note that this equation is scaled by -2, this is to reflect the difference of the maximum (1.0) from the minimum (-1.0) values of each of the normalized response surface equations.

4.4.4   Requirements Grouping

Unfortunately, the current target value formulation still allows redundant requirements to affect the location of the target value. This is seen in equation (44), where the target values are dependent upon the matrix used to recompose the original requirements. Every redundant requirement amongst the originals will skew the target slightly. However, some method used to *group* the requirements together could keep this in check. This grouping

procedure would modify the weights of the individual requirements to reflect redundancy via an additional term. This term would be based on

$$w_j^g = w_j \frac{1}{n_g} \qquad (46)$$

where w_j^g is the modified grouped weight of the j^t requirement and n_g is the number of requirements in the same group as w_j. This ensures that any requirement that is redundant or close to redundant with another does not unnecessarily skew the target values. If any requirement is not in a particular group then w_j^g will be identical to w_j.

The method of grouping requirements is still a subject of research as of this writing. Some techniques are currently being investigated. One promising idea uses vector angles. This technique calculates the vector angle between the *rows* of the $\mathbf{C}^+$ matrix. Each row represents the mapping of the original requirements to the characteristic requirements space; therefore, those that are in the same group will have very small vector angles. The vector angle [Strang, 1988] is found from

$$\theta_{ij} = \cos^{-1}\left(\frac{\vec{c}_i \cdot \vec{c}_j}{\|\vec{c}_i\|\|\vec{c}_j\|}\right) \qquad (47)$$

where θ_{ij} is the vector angle between two requirements and $\vec{c}_i$ refers to the ith *row* vector of $\mathbf{C}^+$. Note that the operator in the numerator is a dot product. The vector angles for the $\mathbf{C}^+$ matrix of the beam problem are presented in Table 10.

Table 10: Vector Angles of Requirements for Beam Design Problem

θ (deg)	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
R_1	0.00	36.73	31.65	115.15	124.30	143.27	31.65	124.30
R_2	36.73	0.00	6.81	85.07	101.15	180.00	6.81	101.15
R_3	31.65	6.81	0.00	91.85	107.90	173.19	0.00	107.90
R_4	115.15	85.07	91.85	0.00	19.94	94.93	91.85	19.94
R_5	124.30	101.15	107.90	19.94	0.00	78.85	107.90	0.00
R_6	143.27	180.00	173.19	94.93	78.85	0.00	173.19	78.85
R_7	31.65	6.81	0.00	91.85	107.90	173.19	0.00	107.90
R_8	124.30	101.15	107.90	19.94	0.00	78.85	107.90	0.00

If the angle is zero or relatively small then the requirements are likely in the same group. These small angles are highlighted in Table 10 and indicate possible requirements groups for this problem. Guidelines for how small angles must be for the requirements to be grouped are the subject of further research. Note that this formulation should not allow a requirement to be part of more than one group, though one can easily imagine a situation where one requirement falls in the middle of two, ungrouped requirements. The final grouping algorithm should have logic to deal with this phenomenon.

Automation of the grouping process, while still under development, seems to be in the realm of pattern recognition. The author has little experience with this field, but "learning" techniques such as Support Vector Machines [Cristianini and Shawe-Taylor, 2000] appear promising. This subject of research is covered in greater detail in the research plan, outlined in the next chapter.

4.5　Decomposition-Based Decision Making

The tools given up to this point form the building blocks of a new process dubbed *Decomposition-Based Decision Making* (DBDM). This process takes the elements of entropy-based static weighting, dynamic weighting with constraint and threshold values, singular value decomposition, and requirements grouping and adds them to the existing generalized MCDM approach given in section 4.1. The entire process then becomes a compromise programming problem of the form of equation (45). For clarity, this is formally written as

$$\min \quad F(\vec{x}) = \left[\sum_{k=1}^{r} \left(\frac{\sigma_k \left(R_k'(\vec{x}) - t_k(\vec{x}) \right)}{-2} \right)^p \right]^{\frac{1}{p}} \tag{48}$$

subject to

$$\vec{x}_{lb} \le \vec{x} \le \vec{x}_{ub}$$

where $\vec{x}_{lb}$ and $\vec{x}_{ub}$ refer to the lower and upper bounds of the design variables, respectively. Equation (48) shows the appropriate quantities that are a function of $\vec{x}$. There are no explicit constraints listed in the optimization problem (other than the bounds on the design variables) because these constraints are handled implicitly through the dynamic weights, which in turn effect the target values $\vec{t}(\vec{x})$. The thresholds also make their mark on the target values through the dynamic weights.

Of pertinent interest in equation (48) is the selection of values for σ_k, the scale factors for the characteristic requirements. These factors represent the importance of one characteristic over another. The choice of these values is still being investigated, though one theory is that the singular values of the characteristics should be used as a scale factor. However, these singular values may be biased by redundant requirements in the original decomposition, so perhaps the best solution is to only use the singular values to choose r (the effective rank of the characteristics) and then weight each of these "important" characteristics evenly. This is discussed more below in the DBDM results for the beam design problem.

The compromise programming problem given by equation (48) yields a deterministic solution. Probabilistic DBDM incorporates probability distributions about the entropy-based static weight values for each of the requirements. This in turn effects the total weight value, ultimately manifesting itself in uncertainty in the target values.

The entire DBDM process is outlined in Figure 27 for greater clarity. It begins as the generalized MCDM problem, utilizing normalized surrogate models, and follows with generation of static weights, enumeration of constraints and thresholds, and singular value decomposition, culminating with compromise programming.

4.5.1　Results for Beam Design Problem

There are several features of DBDM that are illustrated on the beam problem. Furthermore, there are still some pertinent areas of research for the overall process that can be aided with small perturbations to the technique. Namely, these are the choice of number of characteristics to use, the scaling of the characteristics (either equal or by the singular values), and the importance of requirements grouping.

One of the stated benefits of DBDM is the ability to identify and negate the effect of redundant requirements. To illustrate this, eight cases were executed for the beam
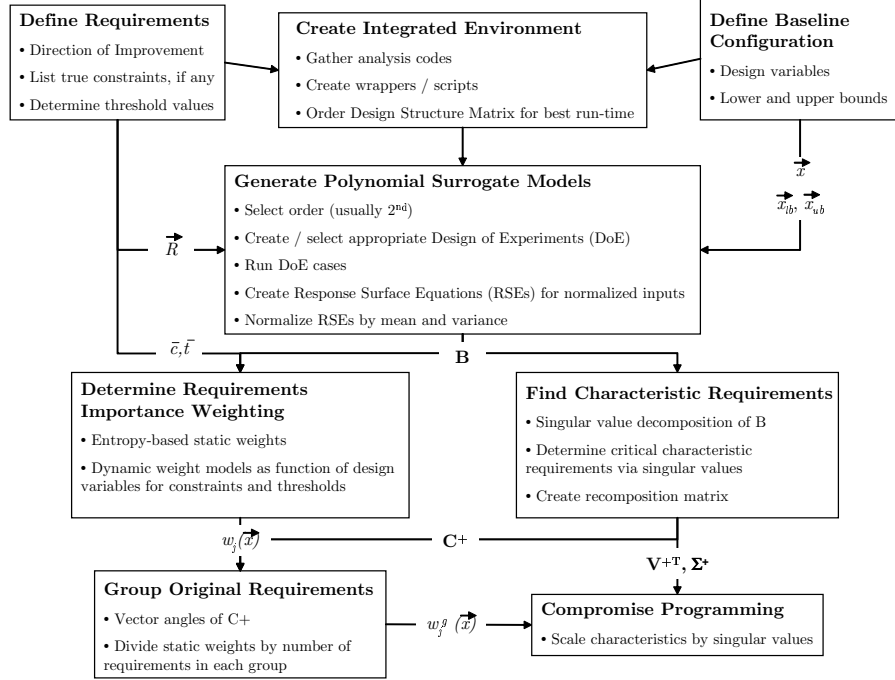
Figure 27: Flowchart for Decomposition-Based Decision Making

design problem. Four cases involved the six-requirement problem, and four with the eight-requirement problem. If DBDM is effective, the differences between the six- and eight-requirement cases should be small, or at least smaller than the differences between the six- and eight-requirement CP problem seen before in Table 3. All eight cases were executed with $r = 3$ indicating that three characteristic requirements were adequate to describe the design space. Four were run with the corresponding singular values as the scale factors for the characteristics (σ_k in equation (48)) and an additional four cases with no scale factors. Table 11 gives the results for the four cases scaled by the singular values and Table 12 gives the unscaled results, each for various numbers of requirements, with and without requirements grouping (as given by equation (46)).

The most striking differences within both tables are the changes between the six- and eight-requirement answers for the ungrouped DBDM problem. This illustrates that some type of requirements grouping is necessary so redundant requirements do not bias the target values given by equation 44. However, little more can be said about the choice of scale factors. The cases scaled with the singular values certainly have different answers than the unscaled results, but which are "better?" The results in Tables 11 and 12 are all expressed with three critical characteristics and provide little guidance. Additional cases were run for the eight-requirement problem with grouping for different numbers of critical characteristics, from two to six. Note that the reality of the problem shows that only five critical characteristics are needed to completely describe the problem, so the addition of a sixth characteristic should have no effect on the answer if the DBDM algorithm should have any merit. The results for the scaled and unscaled DBDM problems for various numbers of critical characteristics are shown in Tables 13 and 14.

These tables show that the problem as scaled by the singular values produces more

Table 11: DBDM Results for Beam Problem Scaled by Singular Values

Case	6 requirements, no grouping	8 requirements, no grouping	6 requirements, grouped	8 requirements, grouped
AR	1.000	1.000	1.000	1.000
w (m)	0.0576	0.0626	0.0569	0.0580
F*	0.5846	0.6705	0.5519	0.5585
R_1	5.570	7.967	5.305	5.767
R_2	51.98	61.41	50.83	52.82
R_3	2.985	3.846	2.886	3.059
R_4 (m)	1.23E-04	1.03E-04	1.26E-04	1.21E-04
R_5 (m)	0.02496	0.01686	0.02624	0.02407
R_6 (kg)	9.276	10.960	9.071	9.426
R_7	N/A	3.846	N/A	3.059
R_8 (m)	N/A	0.01054	N/A	0.01504

Table 12: Unscaled DBDM Results for Beam Problem

Case	6 requirements, no grouping	8 requirements, no grouping	6 requirements, grouped	8 requirements, grouped
AR	1.000	1.000	1.000	1.000
w (m)	0.0566	0.0614	0.0552	0.0567
F*	0.5251	0.4851	0.4886	0.4793
R_1	5.175	7.337	4.625	5.229
R_2	50.25	59.08	47.73	50.49
R_3	2.836	3.626	2.624	2.857
R_4 (m)	1.28E-04	1.08E-04	1.35E-04	1.27E-04
R_5 (m)	0.02691	0.01851	0.03007	0.02663
R_6 (kg)	8.967	10.543	8.519	9.011
R_7	N/A	3.626	N/A	2.857
R_8 (m)	N/A	0.01157	N/A	0.01664

Table 13: DBDM Results for Beam Problem with Varying Critical Characteristics Scaled by Singular Values

Characteristics	2	3	4	5	6
AR	1.000	1.000	1.000	1.000	1.000
w (m)	0.0579	0.0580	0.0580	0.0580	0.0580
F*	0.5317	0.5585	0.5840	0.5840	0.5840
R_1	5.725	5.767	5.767	5.766	5.766
R_2	52.64	52.82	52.82	52.81	52.81
R_3	3.043	3.059	3.059	3.058	3.058
R_4 (m)	1.22E-04	1.21E-04	1.21E-04	1.21E-04	1.21E-04
R_5 (m)	0.02426	0.02407	0.02407	0.02408	0.02408
R_6 (kg)	9.394	9.426	9.426	9.425	9.425
R_7	3.043	3.059	3.059	3.058	3.058
R_8 (m)	0.01516	0.01504	0.01504	0.01505	0.01505

Table 14: Unscaled DBDM Results with Varying Critical Characteristics for Beam Problem

Characteristics	2	3	4	5	6
AR	1.000	1.000	1.955	2.000	2.000
w (m)	0.0577	0.0567	0.0781	0.0814	0.0824
F*	0.2744	0.4793	0.6159	0.7591	0.8849
R_1	5.608	5.229	3.000	3.243	3.355
R_2	52.14	50.49	49.01	52.01	53.29
R_3	2.999	2.857	3.843	4.249	4.404
R_4 (m)	1.23E-04	1.27E-04	1.30E-04	1.22E-04	1.19E-04
R_5 (m)	0.02479	0.02663	0.01289	0.01106	0.01057
R_6 (kg)	9.305	9.011	8.746	9.282	9.510
R_7	2.999	2.857	3.843	4.249	4.404
R_8 (m)	0.01549	0.01664	0.00805	0.00691	0.00660

consistent answers, whereas the unscaled problem changes dramatically as the number of critical characteristics increases. Furthermore, the unscaled problem shows changes at $r = 6$ critical characteristics, which is not physical since the problem only has five linearly independent requirements. Thus, it appears that the scaled problem should be the most accurate formulation.

4.6 Directions for Large-Scale Decision Making

The formulation for Decomposition-Based Decision Making is still evolving as this research progresses, but the results seem to resolve or circumvent the issues associated with generalized MCDM applied to large systems. Preliminary studies using the example beam design problem indicate that the formulation given in equation (48) seems to work best when scaled by the singular values and combined with a requirements grouping technique. Of these, the algorithm used for requirements grouping and the tolerance regarding how "close" requirements must be in order to belong to the same group are a matter of much debate. Still, the initial results are promising for DBDM. The pertinent features include:

- Polynomial surrogate models of the responses normalized by their mean and variance;

- Static weighting values determined by the threshold-modified "entropy" of the decision space;

- Dynamic weighting with the allowance for constraint and threshold values (implicit utility);

- Singular value decomposition of the decision space to identify the characteristic requirements which form the key tradeoffs of the system;

- Ideal tradeoff targets based on the relative contribution of the original requirements to the characteristics;

- Grouping of the original requirements to eliminate double-weighting;

- Compromise programming to find the solution closest to the ideal tradeoff in the characteristic decision space scaled by the singular values of these characteristics.

These features should enable a decision maker to determine the best tradeoff solution for a large-scale multicriteria problem. Ultimately, this method can be made probabilistic to handle uncertainty in the requirements. This uncertainty can be manifested with probability distributions about the static weight values, which in turn cascade into compromise programming problem via the target values. The solution that is most robust with respect to changes in the importance of the metrics will in turn be the solution best suited to uncertain requirements.

Bibliography

AIAA Technical Activities Committee. Current state of the art on multidisciplinary design optimization (MDO). AIAA White Paper, 1991.

Asimow, Morris. *Introduction to Design.* Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

Baker, Andrew P. *The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design.* Ph.D. thesis, Georgia Institute of Technology, Atlanta, Georgia, 2002.

Baker, Andrew P. and Mavris, Dimitri N. Assessing the simultaneous impact of requirements, vehicle characteristics, and technologies during aircraft design. 2001-0533. 39th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2001.

Bandte, Oliver. *A Probabilistic Multi-Criteria Decision Making Technique for Conceptual and Preliminary Aerospace Systems Design.* Ph.D. thesis, Georgia Institute of Technology, Atlanta, Georgia, 2000.

Baugher, Jay. USAAC / USAAF / USAF bomber aircraft. 2005. http://home.att.net/ j̃baugher/usafbombers.html.

Borer, Nicholas K. and Mavris, Dimitri N. Requirements exploration for a notional multi-role fighter. 2003-6709. 3rd Aviation Technology, Integration, and Operations Technical Forum, American Institute of Aeronautics and Astronautics, 2003.

Borer, Nicholas K. and Mavris, Dimitri N. Formulation of a multi-mission sizing methodology for competing configurations. 2004-0535. 42nd Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2004.

Brassard, Michael and Ritter, Diane. *The Memory Jogger II.* GOAL / QPC, Salem, New Hampshire, 1994.

Braun, Robert, Gage, Peter, and Kroo, Ilan. Implementation and performance issues in collaborative optimization. 96-4017. 6th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, 1996.

Carty, A. An approach to multidisciplinary design, analysis, and optimization for rapid conceptual design. 2002-5438. 9th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, Atlanta, Georgia, 2002.

Chvátal, Vasek. *Linear Programming.* W. H. Freeman and Company, New York, New York, 1983.

Cristianini, Nello and Shawe-Taylor, John. *An Introduction to Support Vector Machnies and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge, United Kingdom, 2000.

Czysz, P. A., Dighton, R. D., and W. P. Murden, Jr. Designing for air superiority. Technical Report 73-0800, McDonnell Aircraft Company, 1973.

Deb, Kalyanmoy. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley and Sons, Chichester, England, 2001.

DeLaurentis, D., Mavris, D. N., and Schrage, D. P. System synthesis in preliminary aircraft design using statistical methods. 20th Congress of the International Council of the Aeronautical Sciences, International Council of the Aeronautical Sciences, Sorrento, Italy, 1996.

Department of the Navy. Navy historical center, naval aviation history branch. 2005. http://www.history.navy.mil/branches/nhcorg4.htm.

Dieter, George E. *Engineering Design: A Meterials and Processing Approach, Third Edition.* McGraw-Hill, Inc., Boston, Massachusetts, 2000.

Eckels, W. E. Civil transport aircraft design methodology. 83-2463. Aircraft Design, Systems and Technology Meeting, American Institute of Aeronautics and Astronautics, Fort Worth, Texas, 1983.

Engineous Software. iSIGHT - integrate, automate, and optimize your manual design processes. 2005. http://www.engineous.com/product_iSIGHT.htm.

Federal Aviation Administration. Code of federal regulations, title 14, part 25: Airworthiness standards: Transport category airplanes. 2005. http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?&c=ecfr&tpl=/ecfrbrowse/Title14/14tab _02.tpl.

Federation of American Scientists. U.S. military aircraft. 2005. http://www.fas.org/man/dod-101/sys/ac/index.html.

Fox, Eric P. The Pratt and Whitney probabilistic design system. 94-1442. 35th Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 1994.

Gerhards, Roland, Meller, Frank, Dirks, Gregor A., and Szodruch, Joachim. Aircraft design optimizing operators, environmental system and manufacturers requirements. 2000-01-5546. 2000 World Aviation Congress, Society of Automotive Engineers, 2000.

Greenway, M. K. and Koob, S. J. Relating take-off-gross-weight and mission requirements to geometrically optimized aircraft. 78-0098. 16th Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics, 1978.

Guynn, M. D. and Olson, E. D. Evaluation of an aircraft concept with over-wing, hydrogen-fueled engines for reduced noise and emissions. Technical Report TM-2002-211926, National Aeronautics and Space Administration, Hampton, Virginia, 2002.

Hibbeler, R. C. *Mechanics of Materials, Third Edition.* Prentice Hall, Upper Saddle River, New Jersey, 1997.

Hwang, Ching-Lai and Masud, Abu Syed Md. *Multiple Objective Decision Making: Methods and Applications.* Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin, Germany, 1979.

Hwang, Ching-Lai and Yoon, Kwangsun. *Multiple Attirbute Decision Making: Methods and Applications*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, Berlin, Germany, 1981.

Ismail-Yahaya, Amir and Messac, Achille. Effective generation of the pareto frontier: The normalized normal constraint method. 2002-1232. 43rd Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 2002.

James, M. L., Smith, G. M., Wolford, J. C., and Whaley, P. W. *Vibration of Mechanical and Structural Systems: With Microcomputer Applications, Second Edition*. HarperCollins College Publishers, New York, New York, 1994.

Joy, D. P. and Simonds, R. M. The R_f graphical method of parametric analysis for the development of optimum preliminary design aircraft. Technical Report 107, Advanced Research Division of Hiller Helicopters, 1956.

Kam, J. Vander and Gage, P. The launch vehicle language (LVL) data model for evaluating reusable launch vehicle concepts. 2003-1330. 41st Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, Reno, Nevada, 2003.

Kirby, M. R. and Mavris, D. N. Forecasting technology uncertainty in preliminary aircraft design. 1999-01-5631. 1999 World Aviation Conference, Society of Automotive Engineers, San Francisco, California, 1999.

Kirby, M. R. and Mavris, D. N. An approach for the intelligent assessment of future technology portfolios. 2002-0515. 40th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, Reno, Nevada, 2002.

Ledsinger, L. A. and Olds, J. R. Multidisciplinary design optimization techniques for branching trajectories. 98-4713. 7th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, 1998.

Leonard, John. *Systems Engineering Fundamentals*. Defense Systems Management College, Fort Belvoir, Virginia, 1999.

Lin, R. and Afjeh, A. A. An extensible, interchangeable and sharable database model for improving multidisciplinary aircraft design. 2002-5613. 9th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, Atlanta, Georgia, 2002.

Mattingly, Jack D., Heiser, William H., and Pratt, David T. *Aircraft Engine Design, Second Edition*. American Institute of Aeronautics and Astronautics, Reston, Virginia, 2002.

Mavris, D. N. and Kirby, M. R. Technology identification, evaluation, and selection for commercial transport aircraft. 2456. 58th Annual Conference of Society of Allied Weight Engineers, Society of Allied Weight Engineers, San Jose, California, 1999.

Mavris, D. N., Mantis, G. C., and Kirby, M. R. Demonstration of a probabilistic technique for the determination of aircraft economic viability. 975585. 2nd World Aviation Congress and Exposition, Society of Automotive Engineers, Anaheim, California, 1997.

Mavris, Dimitri N. and Bandte, Oliver. Economic uncertainty assessment using a combined design of experiments / monte carlo simulation approach with application to an hsct. 17th Annual Conference of the International Society of Parametric Analysts, International Society of Parametric Analysts, 1995.

Mavris, Dimitri N. and Bandte, Oliver. Comparison of two probabilistic techniques for the assessment of economic uncertainty. 19th Annual Conference of the International Society of Parametric Analysts, International Society of Parametric Analysts, 1997.

Mavris, Dimitri N. and Borer, Nicholas K. Development of a multi-mission sizing methodology applied to the common support aircraft. 2001-01-3014. World Aviation Congress and Exposition, Society of Automotive Engineers, 2001.

Mavris, Dimitri N. and DeLaurentis, Deniel. Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design. 145.1. ICAS, 2000.

McCullers, L. A. Aircraft configuration optimization including optimized flight profiles. In Sobieski, J., editor, *Proceedings of the Symposium on Recent Experiences in Multidisciplinary Analysis and Optimization*, CP-2327, pages 396–412. National Aeronautics and Space Administration, 1984.

McCullers, L. A. *Flight Optimization System Release 5.94 User's Guide*. National Aeronautics and Space Administration, 5.94 edition, 1998.

McDonnell-Douglas Corporation. *F/A-18C Substantiating Performance Data with F404-GE-402 Engines*. McDonnell-Douglas Corporation, MDC 91B0290 edition, 1996.

MIT Axiomatic Design Group. General axiomatic design concepts. 2005. http://web.mit.edu/axiom/www/introduction.shtml.

Mullur, Anoop A., Mattson, Christopher A., and Messac, Achille. Pitfalls of the typical construction of decision matrices for concept selection. 2003-0466. 41st Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 2003.

Myklebust, Arvid and Gelhausen, Paul. Putting the ACSYNT on aircraft design. *Aerospace America*, pages 26–30, 1994.

Nakos, George and Joyner, David. *Linear Algebra with Applications*. Brooks/Cole, Pacific Grove, California, 1998.

Naval Air Systems Command. *Standard Aircraft Characteristics: F-14A Tomcat*. Naval Air Systems Command, NAVAIR 00-110AF14-1 edition, 1976.

Naval Air Systems Command. *Standard Aircraft Characteristics: Navy Model A-6E (TRAM) Aircraft*. Naval Air Systems Command, NAVAIR 00-110AA6-5 edition, 1984.

Naval Air Systems Command. *Standard Aircraft Characteristics: F/A-18C Hornet, (2) F404-GE-402 Engines*. Naval Air Systems Command, NAVAIR 00-110AF18-4 edition, 1996.

Nelson, J. M., Kawai, R. T., and Gregg, R. D. A study of the economic benefit potential of internodal transports. Technical Report CR-2001-210847, National Aeronautics and Space Administration, Hampton, Virginia, 2001.

Neter, John, Kutner, Michael H., Nactsheim, Chistopher J., and Wasserman, William. *Applied Linear Statistical Models, Fourth Edition.* McGraw-Hill, Boston, Massachusetts, 1996.

Olds, J. System sensitivity analysis applied to the conceptual design of a dual-fuel rocket SSTO. 94-4339. 5th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, 1994.

Pareto, Vilredo. *Manual of Political Economy.* Augustus M. Kelley, New York, New York, 1971.

Parker, James L. Mission requirements and aircraft sizing. 86-2622. Aircraft Systems, Design, and Technology Meeting, American Institute of Aeronautics and Astronautics, 1986.

Phoenix Integration. Phoenix Integration Products: ModelCenter. 2005. http://www.phoenix-int.com/products/ModelCenter.php.

Piccirillo, Albert C. Origins of the F-22 raptor. 98-5566. Society of Automotive Engineers, 1998.

Pugliese, J. J. Design synthesis. The Aircraft Design Process Seminar, Society of Aeronautical Weight Engineers, Arlington, Texas, 1971.

Raymer, D. P. Vehicle scaling laws for multidisciplinary optimization: Use of net design volume to improve optimization realism. 2001-5246. 1st Aerospace Technology, Integration, and Operations Forum, American Institute of Aeronautics and Astronautics, Los Angeles, California, 2001.

Raymer, Daniel P. *Aircraft Design: A Conceptual Approach, Third Edition.* American Institute of Aeronautics and Astronautics, Reston, Virginia, 1999.

Rich, Ben R. and Janos, Leo. *Skunk Works.* Little, Brown and Company, Boston, Massachusetts, 1994.

Roskam, Jan. *Airplane Design Part I: Preliminary Sizing of Airplanes.* Roskam Aviation and Engineering Corporation, Ottawa, Kansas, 1989.

SAS Institute. JMP Software - Data Analysis - Statistics - Six Sigma - DOE. 2005. http://www.jmp.com/.

Schrage, D. P. Technology for rotorcraft affordability through integrated product/process development (IPPD). American Helicopter Society 55th Annual Forum, American Helicopter Society, Montreal, Canada, 1999.

Shepard, Roger N. On subjectively optimum selection among multiattribute alternatives. In Shelly, Maynard W. and Bryan, Glenn L., editors, *Human Judgements and Optimality*, chapter 14. John Wiley and Sons, Inc., New York, New York, 1964.

Sherman, Robert and Hardiman, Matthew X. F-35 Joint Strike Fighter. 2003. http://www.fas.org/man/dod-101/sys/ac/f-35.htm.

Simonds, R. M. A generalized graphical method of minimum gross weight estimation. 135. The National Conference of the Society of Aeronautical Weight Engineers, Society of Aeronautical Weight Engineers, San Diego, California, 1956.

Simos, D. and Jenkinson, L.R. Optimisation of the conceptual design and mission profiles of short-haul aircraft. 86-2696. Aircraft Systems, Design, and Technology Meeting, American Institute of Aeronautics and Astronautics, Dayton, Ohio, 1986.

Sobieszczanski-Sobieski, Jaroslaw. Sensitivity analysis and multidisciplinary optimization for aircraft design: Recent advances and results. *Journal of Aircraft*, 27(12):993–1001, 1990.

Sobieszczanski-Sobieski, Jaroslaw, Agte, Jeremy S., and Sandusky, Robert R. Bi-level integrated system synthesis (BLISS). 98-4916. American Institute of Aeronautics and Astronautics, 1998.

Strang, Gilbert. *Linear Algebra and its Applications, Third Edition.* Brooks/Cole, Pacific Grove, California, 1988.

Suh, Nam P. *The Principles of Design.* Oxford University Press, New York, New York, 1990.

The Mathworks, Inc. Matlab - the language of technical computing. 2005. http://www.mathworks.com/products/matlab/.

Toppan, Andrew. World navies today: U.S. navy carrier air wings. 2003. http://www.hazegray.org/worldnav/usa/wings.htm.

Torenbeek, Egbert. *Synthesis of Subsonic Airplane Design.* Delft University Press, Delft, Holland, 1982.

Triantaphyllou, Evangelos. *Multi-Criteria Decision Making Methods: A Comparative Study.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

Vanderplaats, Garret N. *Numerical Optimization Techniques for Engineering Design, Third Edition.* Vanderplaats Research and Development, Inc., Colorado Springs, Colorado, 1999.

Weisstein, Eric W. Heaviside step function. 2005. http://mathworld.wolfram.com/ HeavisideStepFunction.html.

Yoon, K. Paul and Hwang, Ching-Lai. *Multiple Attribute Decision Making: An Introduction.* Sage Publications, Thousand Oaks, California, 1995.

Young, James A., Anderson, Ronald D., and Yurkovich, Rudolph N. A description of the F/A-18E/F design and design process. 98-4701. American Institute of Aeronautics and Astronautics, 1998.

Young, Warren C. and Budynas, Richard G. *Roark's Formulas for Stress and Strain.* McGraw-Hill, Boston, Massachusetts, 7th edition, 2001.

Zeleny, Milan. *Multiple Criteria Decision Making.* McGraw-Hill, New York, New York, 1982.